Data processing handover in the multi-access edge computing setting

Guillermo Alonso Núñez

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 29.7.2019

Supervisor

Prof. Dr. Tech. Antti Ylä-Jääski

Advisor

Dr. Kimmo Hätönen

Aalto University
School of Science

Aalto University
School of Science

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Guillermo Alonso Núñez

Title Data processing handover in the multi-access edge computing setting

Degree programme ICT Innovation

Major Data Science **Code of major** SCI3095

Supervisor Prof. Dr. Tech. Antti Ylä-Jääski

Advisor Dr. Kimmo Hätönen

Date 29.7.2019 **Number of pages** 57 **Language** English

Abstract

The multi-access edge computing (MEC) technology is a key pillar of the 5th generation (5G) telecommunication network. Among other benefits, it will allow for ultra-low latency communications by bringing computations closer to the user equipment (UE). However, when the UE changes its position, the problem of keeping computations close to it arises. In this work I study this problem related to handover, taking the Megasense project as a real life use case. I propose, implement and analyze a solution that aims at solving the aforementioned problem within a prototype system developed by Nokia Bell Labs.

Keywords multi-access edge computing, 5g, handover, mobile user equipment

Preface

I want to thank my family, especially my mother, for supporting me during my studies.

I also want to thank my advisor from Nokia Bell Labs Kimmo Hätönen for his guidance and feedback on my work for this thesis, and my supervisor from Aalto University Antti Ylä-Jääski for allowing me to develop my thesis in such an interesting area.

Espoo, 29.7.2019

Guillermo Alonso Núñez

Contents

Abstract 3

Preface 4

Contents 5

Abbreviations and acronyms 7

1 Introduction 9
- 1.1 Problem statement . . . 10
- 1.2 Objectives . . . 10
- 1.3 Structure . . . 10

2 Background 11
- 2.1 Cellular network architecture . . . 11
  - 2.1.1 Standardization . . . 12
  - 2.1.2 Current cellular network architecture . . . 13
- 2.2 5G . . . 13
  - 2.2.1 5G adoption . . . 14
  - 2.2.2 5G user equipment mobility . . . 15
  - 2.2.3 Network slicing . . . 15
  - 2.2.4 Multi-access edge computing . . . 15
  - 2.2.5 Complexity of the 5G cellular network . . . 16
- 2.3 Nokia's proof of concept system . . . 17
  - 2.3.1 Service architecture . . . 18
  - 2.3.2 Publish/subscribe pattern . . . 19
  - 2.3.3 Main components of the system . . . 19
- 2.4 Megasense project . . . 21
  - 2.4.1 Air pollution . . . 21
  - 2.4.2 Scope and goals . . . 22

3 Data processing handover and solutions 23
- 3.1 Rationale . . . 23
  - 3.1.1 The handover problem . . . 23
  - 3.1.2 Problem definition . . . 24
- 3.2 State of the art . . . 25
  - 3.2.1 Handover and 5G . . . 25
  - 3.2.2 Handover and self-driving cars . . . 27
- 3.3 Methodology . . . 30
  - 3.3.1 Break-before-make handover . . . 30
  - 3.3.2 Make-before-break handover . . . 34
- 3.4 Enhanced make-before-break handover . . . 36
- 3.5 Implementation . . . 37

4 Evaluation **38**
4.1 Testing environment 38
4.2 Overhead of spawning data fetcher 39
4.3 Results 39
4.3.1 Stream managing overhead 39
4.3.2 Other differences 45
4.4 Conclusion 46

5 Further work **47**
5.1 Stream synchronization 47
5.2 More realistic implementation and testing 49
5.3 Resource cleanup after handover 49

6 Summary **51**

References **52**

Abbreviations and acronyms

2G	second generation
3G	third generation
3GPP	3rd generation partnership project
4G	fourth generation
5G	fifth generation
AI	artificial intelligence
AR	augmented reality
BbM	break-before-make
BS	base station
BTS	base transceiver station
CAM	cooperative awareness message
CDF	cumulative distribution function
CN	core network
D2D	device-to-device
DENM	decentralized environmental notification messages
DF	data fetcher
DH	data hub
DS	data switch
EC	edge cloud
eNB	evolved node b
ETSI	european telecommunications standards institute
gNB	next generation node b
HO	handover
IDE	integrated development environment
IT	information technology
ITS	intelligent transportation system
IoT	internet of things
KPI	key performance indicator
LTE	long term evolution
MbB	make-before-break
MEC	multi-access edge computing
ML	machine learning
NB	node b
NB-IoT	narrowband internet of things
PM	particulate matter
PUB/SUB	publish/subscribe
QoS	quality of service
RA	random access
RAN	radio access network
RLF	radio link failure
RRM	radio resource management

SMEAR	stations measuring earth surfaces and atmosphere relations
UE	user equipment
UEI	user equipment interface
URLLC	ultra-reliable low-latency communications
V2I	vehicle-to-infrastructure
V2P	vehicle-to-pedestrian
V2V	vehicle-to-vehicle
V2X	vehicle-to-everything
VANET	vehicular ad-hoc network

1 Introduction

In August 2018, the estimated number of internet of things (IoT) devices deployed worldwide was 7 billion. This number is expected to grow in the coming years, reaching 20 billion by 2025 [35]. This indicates that the amount of data that needs to be preprocessed, transferred and analyzed is growing at exponential rates. As a result, telecommunication operators need to constantly evolve and adapt to meet the increasing demands of their users. They need to develop new ways of handling and exploiting the vast amount of devices that are and will be connected to the network.

The next milestone for telecommunication networks is known as fifth generation (5G) and it is planned to be deployed commercially in most markets by 2020 [8]. Its main characteristics are high data transferring rates, ultra-low latency and massive device connectivity. These provide the perfect conditions for supporting the IoT era, among others. However, the adoption process will be slow since 5G will require deploying a dense and large-scale infrastructure [9].

In the past, companies owned their high-end servers. However, the trend shifted in recent times due to the advent of cloud computing. Cloud computing delegates data and its processing to large dedicated data centers somewhere on the internet [6]. This allows companies to save costs and focus on business opportunities [7]. Nowadays, the trend is starting to shift again, due to the new business opportunities that 5G enables. Recent studies show that cloud computing will not achieve acceptable quality of service (QoS) levels for some of the applications that 5G and multi-access edge computing (MEC) technologies enable, such as those requiring ultra-low latency in the millisecond scale [34]. MEC technologies provide local computing and storage capabilities at the edge cloud (EC) level, close to the user equipment (UE).

For many applications, such as self-driving cars, it is crucial that the outcome of the data analysis is available in almost real-time, with delays of milliseconds. Delays of seconds can mean the difference between life and death. To achieve this latency, the analysis of data must be carried geographically close to its data source, avoiding having to send the data across the network to some remote data center. This is where MEC comes into play. However, in the MEC setting, computing and storage resources at the EC will be limited. This contrasts with the cloud computing approach, in which resources are usually thought as unlimited due to dynamic resource allocation and scaling. For this reason, it becomes crucial to decide which data needs immediate analysis and which data can be sent to the cloud servers for further analysis.

A project that will benefit from the 5G and MEC setting is the Megasense project. The Megasense project addresses the global challenge of pollution modeling and prediction while considering the limitations of the state of the art: low density of measurement stations and lack of high-resolution spatial-temporal data [57]. Currently, the project relies on stationary sensors that generate data that is then analyzed in a cloud server [56]. This will run into performance issues as the number of sensors and collected data points increases. Incorporating mobile UE in the form of sensors attached to buses, cars or even clothing apparel, and the need to provide results in almost real-time will further aggravate the problem.

In this thesis, I study the problem of data processing handover (HO) in the MEC

setting and take the Megasense project as a use case. I define the data processing HO as the task of keeping the computations about some data geographically close to the sensors, as the UE to which the sensors are attached roams and connects to different ECs. I take the Megasense project as a study case. The Megasense project will incorporate mobile UE for data collection. The data centers present at each EC must be provided upon request with instructions about how to interpret the data coming from the different types of sensors. Each sensor model will have its own set of instructions. Since the number of sensors is not bounded and resources are limited, it is unfeasible to store instructions for every sensor at the EC level. Instead, the instructions will need to be made available upon request. For this reason, the Megasense project is a suitable testbed for the solution developed in this thesis. The work presented in this thesis is implemented on a proof of concept system developed by Nokia Bell Labs. The solution is quantitatively evaluated from different aspects.

1.1 Problem statement

In this thesis I aim at answering the following research questions related to the MEC setting:

- How can we optimize Nokia's system at the EC level for allowing an efficient data processing HO solution?
- How can we minimize delays and bandwidth usage when tackling the data processing HO problem within Nokia's system?

1.2 Objectives

The goal of this thesis is to design and implement a solution that allows for data processing HO to happen within Nokia's system. Furthermore, the solution must be thought for the 5G cellular network. In particular, the solution will implement a make-before-break approach regarding connectivity to the different ECs. After the solution is implemented, the results will be measured, evaluated and discussed.

1.3 Structure

The rest of this thesis is organized as follows. Chapter 2 provides a snapshot of the current environment and concepts required to better understand the scope of this thesis. Chapter 3 explains the rationale and development of the developed implementation. Chapter 4 contains the evaluation of the implementation and the gathered results. Chapter 5 points areas of further research. Chapter 6 concludes the thesis by revisiting its main contributions.

2 Background

This section explains the main concepts required to understand the proposed research problem.

2.1 Cellular network architecture

A cellular network or mobile network is a type of communication network in which the link to the end mobile devices is wireless [36]. Cellular networks are divided into generations according to their characteristics. Regardless of their generation, they all share a common high-level architecture, depicted in Figure 1.

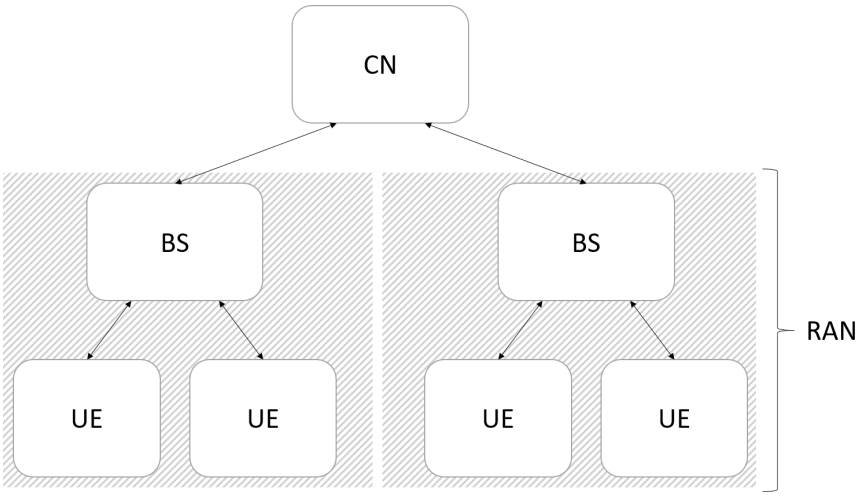

Figure 1: High-level view of the architecture of cellular networks.

The devices that allow for the last wireless link connectivity are called base stations (BSs). BSs are connected to the core network (CN) through some link, such as optic fiber. Devices that connect to the CN through the BSs are called user equipment (UE). Examples of UE include internet of things (IoT) sensors, smartphones and autonomous vehicles. The wireless connection between the UE and the BSs is done through the radio access network (RAN).

To improve the quality of service (QoS) for the users, a single geographical area or cell is usually covered by more than one BS, as depicted in Figure 2. Traditionally, the cell layout is represented as a hexagonal grid, each hexagon representing a unique cell. A cell is typically covered by 3 BSs located in non-neighboring vertices of the hexagon. BSs are equipped with a set of directional antennas aimed in three different directions. This allows for transmitting and receiving data at different frequencies for each neighboring cell. The rationale behind this design is to reuse frequencies on non-neighboring cells, making it possible to cover large areas with a limited set of frequencies [39]. For the rest of this thesis, I will refer to the set of all cells as the cellular network grid, or simply grid.

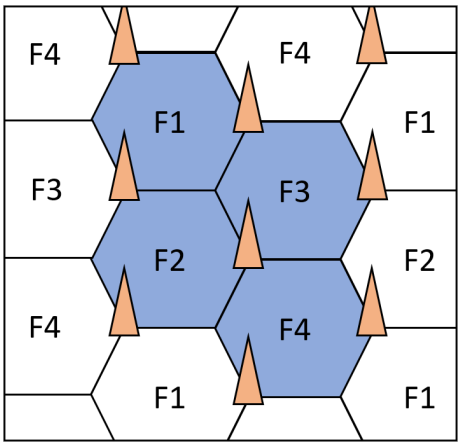

Figure 2: Example of a frequency-reusing pattern for a cellular network. Orange triangles represent BSs. Hexagons in blue highlight a pattern which yields a tessellation of the plane, while ensuring that no pair of neighboring cells use the same frequency.

2.1.1 Standardization

There are several bodies who work on the standardization of the different cellular network generations. Perhaps the most relevant one is the 3rd Generation Partnership Project (3GPP). The 3GPP unites seven telecommunication standard development organizations and provides them with a stable environment to produce the specifications that define 3GPP technologies. The 3GPP covers cellular telecommunication network technologies and provides complete system specifications [48].

The 3GPP was born in 1998 to produce technical specifications and reports for the third generation (3G) mobile system. Since then, it has worked on the development and maintenance of many standards, including the fourth generation (4G) telecommunications network, 4G LTE (4G long term evolution, an advanced version of 4G) and fifth generation (5G) standards.

Table 1 summarizes some of the most relevant differences between generations for this thesis. Note that all the cellular network generations from the second generation (2G) onwards share the same conceptual architecture, depicted in Figure 1.

Generation	Year	Base Station acronym	Base Station coverage radius	Latency	Bandwidth
2G	1991	BTS	2-5 km (urban) 10-32 km (rural)	300 ms	50 Kbps
3G	1998	NB	2-5 km (urban) 10-32 km (rural)	120 ms	3.1 Mbps
4G	2008	eNB	2-5 km (urban) 10-32 km (rural)	60 ms	100-300 Mbps
5G	2018	gNB	250-300 m	1 ms (*)	10 Gbps (*)

Table 1: Comparison between different cellular network generations [1] [40] [41] [42] [43] [10]. Fields marked with (*) indicate theoretical results.

The concept of BS has received various names across the different generations. In 2G, it was referred to as base transceiver station (BTS). In 3G, it was named Node B (NB). In 4G, it was named Evolved Node B (eNB). In 5G, it is known as Next Generation Node B (gNB). For the rest of this thesis, I will work on and refer to the 5G setting and hence use its wording unless explicitly stated otherwise.

2.1.2 Current cellular network architecture

The current cellular network is highly heterogeneous. Many technologies and infrastructure from different generations of the cellular network coexist. This is partly due to cellular operators being reluctant to get rid of old technologies and infrastructure due to sunk costs and user retention [11]. For instance, some european carriers will not switch off their 2G networks until 2025 [49].

According to data from a collaborative effort to compile information about the coverage of the different cellular network generations, most countries around the world provide some sort of 4G cellular coverage [51]. However, some of them, especially in Africa, only have access to older generations, namely 3G or even 2G technologies. Many countries are already providing access to 5G cellular networks, even if only over some specific areas or as a part of experimental deployments.

2.2 5G

The next generation of cellular networks is known as 5G. As mentioned in section 2.1.1, its standard is being developed by the 3GPP [48]. 5G enables support for ultra-low-cost IoT, support for 10-100 times more UE, latency below 1 ms, ultra-reliable low-latency communications (URLLC), support for 10.000 times more network traffic and data rates over 10 Gbps, among others [28].

These improvements allow the 5G cellular network to provide a large range of services, including IoT-based systems, critical services requiring URLLC and applications requiring more bandwidth than achievable with previous generations. Among the new business opportunities that 5G will enable, we find intelligent transportation systems (ITSs), smart cities, augmented reality (AR) applications, or services related to optimizing processes in the health and industrial sectors [69].

To achieve the 5G goals, a new different set of technologies needs to be in place. Those technologies include the standardization of a new radio spectrum and the research and development of new antenna technologies. A new large-scale infrastructure also needs to be deployed [28].

In contrast with traditional telecommunication networks perspective, 5G not only aims at providing macro-grids covering large geographical areas, but also at providing service for hotspots [9]. Hotspots are small areas that are expected to take on a huge workload during a specific time. Examples of hotspots are concert venues or stadiums. The use of 5G for this use case brings improved QoS when compared to WiFi. Furthermore, 5G considers cells of different sizes and characteristics [38], such as pico cells, with coverage ranges in the tens of meters and thought to be deployed in indoors facilities like shopping malls or train stations. Other types are

femto cells, to be deployed in home and business environments with similar ranges to the pico cells, and micro cells, to fill any possible gaps in coverage in urban areas, with coverage ranges in the hundreds of meters.

However, all the new possibilities 5G will open, come along with greater technical challenges and more complex solutions. An example of such a complex solution is the network-slicing virtualization technology, which I explain later in this chapter.

2.2.1 5G adoption

Despite the promising features of 5G, its adoption will be a slow and gradual process. Existing eNB stations can be upgraded to support 5G capabilities while providing approximately the same coverage [28]. This type of deployment is known as non-standalone. In contrast, the scenario in which only one radio access technology is employed is referred to as standalone deployment [37].

As we can observe in Table 1, the gNB coverage radius is significantly smaller than that of its predecessor, the eNB. This is because 5G employs higher frequency signals to reach greater data rates, among others. As we know, and can appreciate in Figure 3, the frequency of a wave is inversely proportional to its wavelength. Since 5G signals employ higher frequencies, they will also have shorter wavelengths. Hence, they will be more prone to losing energy due to collisions with particles present in the air. This can be seen in Figure 3 by realizing that the total distance wave a) traverses is much larger than that traversed by wave b).

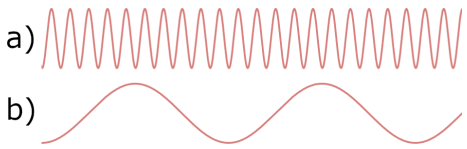

Figure 3: Waves with different characteristics. a) High frequency, short wavelength. b) Low frequency, long wavelength.

Consequently, the number of gNBs that need to be deployed to provide coverage to the same area will be higher. This along with high infrastructure deployment costs detriment the adoption rate of the 5G network. Furthermore, operators are seeking to avoid the 4G rollout experience, in which they struggled to generate additional revenues from the end-users, and are exploring new ways to monetize their investments [52].

The first 5G networks open to the public were available in early April 2019 in South Korea [68] and USA [66]. As of June 2019, the total number of users of 5G networks was estimated to be 10 million [44]. South Korea, a country with 51 million population, broke the million subscribers mark for 5G networks mid-June 2019 [64].

On a global level, as of April 2019, 224 operators in 88 countries were reported as investing in 5G cellular networks either through tests, trials, pilots, planned and/or actual deployments. In particular, 39 of them had announced that they had deployed some form of 5G technology in their networks [67]. It is expected that the first countries hitting a 50% adoption rate of 5G, meaning 50% or more devices connected

to the 5G network, will do so by 2021. However, most markets are expected to hit that mark years later, by 2023-2024.

2.2.2 5G user equipment mobility

The concept of user equipment (UE) mobility has been an active area of research since the first cellular networks were commercially deployed almost 30 years ago [1]. For 5G, new mobility solutions need to be in place to enable low-latency and highly reliable connections. The aim of 5G is to achieve no interruption during mobility, which inevitably requires make-before-break connectivity approaches [30].

For the scope of this thesis regarding mobility, 5G demands low-latency, high-reliability and zero handover (HO) execution time. These parameters were also defined for 4G LTE, but with more relaxed requirements, due to aiming at different use cases [8].

2.2.3 Network slicing

Let us consider different use cases targeted by the 5G cellular network, such as AR applications, smart health care through personal health devices and self-driving cars. Each of these use cases will have its own requirements regarding latency, throughput, capacity and availability [33].

Network slicing is a technology that enables multiple virtual networks to run independently on the same physical network [28]. In other words, it enables for the creation of independent virtual networks aimed at serving different services, with the benefit of only having to deploy one physical network instead of one per service [29]. These virtual networks are also referred to as slices, hence the name network slicing.

The goal of 5G network slicing is to ensure that the requirements are met, from both the application and the customer's perspective. Network slicing is mostly enabled by virtualization and cloud technologies [29]. It is worth mentioning that the 5G standard supports simultaneous UE connections to different network slices [28].

Nowadays, network slicing is still being standardized. While looking promising for the industry, there are still some questions that need to be answered, such as what revenue can be generated exactly from it or what are the consequences of network slicing in operational and investment costs [29].

2.2.4 Multi-access edge computing

One of the benefits of the 5G cellular network will come from the adoption of the multi-access edge computing (MEC) paradigm. MEC offers an information technology (IT) service environment and cloud computing capabilities close to the UE [32].

In the MEC setting, I define an edge or edge cloud (EC) as the RAN and common resources accessible through a set of gNBs. These resources include computing and storage capabilities [5].

As an example, consider a street in which there is a gNB deployed on top of every light pole. A self-driving car that traverses the street will sequentially establish

connections with all the gNBs in it. Suppose that all those gNBs are connected to a certain small local data center providing limited resources. In this scenario, analysis of the most critical data generated by the car can be done very close to the car, with latency in the millisecond scale, enabling the car to make decisions in almost real-time. The rest of the data can be sent to remote data centers with more computing power.

However, the self-driving car will eventually leave the EC it was traversing and enter a new EC with a different set of resources whose access will be provided by a different set of gNBs. When this happens, the network must be able to automatically move the calculations required by the car to the new EC, so that the latency can be kept as low as possible in a transparent manner for the end-point application.

When compared to cloud computing approaches, the main benefits of the MEC paradigm are lower latency, bandwidth usage reduction on the CN and reduced computational workload on the centralized cloud servers. For the 5G network, each EC is expected to support around 100 gNBs and 100,000 UE [30].

Figure 4 depicts a high-level structural view of the MEC setting from the cellular network perspective. There are two things to note in it. The first is that an EC can be accessed through more than one gNB, and the second is that a UE might establish simultaneous connections to different gNBs, regardless of whether they provide access to the same EC or not.

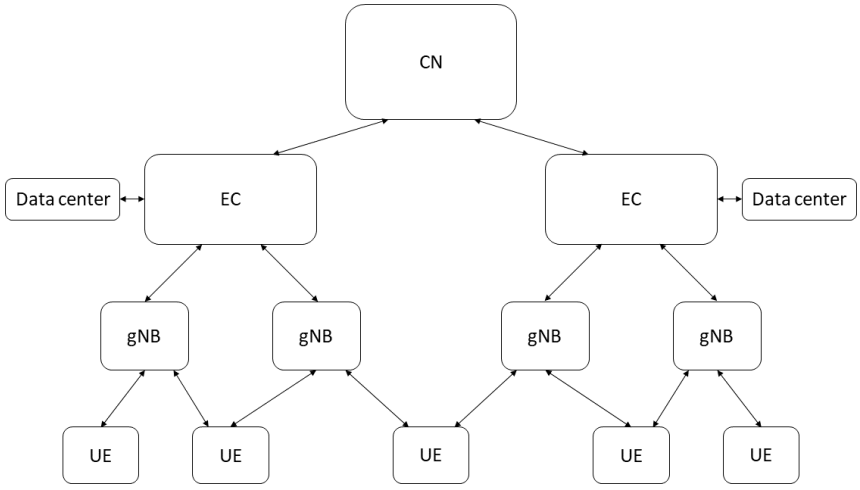

Figure 4: MEC setting high-level structure representation.

2.2.5 Complexity of the 5G cellular network

The 5G networks are more complex than those cellular networks from previous generations. This complexity is due to different aspects. For example, the 5G radio access technology, known as new radio and operating at higher frequencies, introduces more complex antenna configurations and more complex connectivity mechanisms, such as beamforming [28]. Beamforming is a technology alternative to the omnidirectional antennas deployed in previous cellular network generations. Beamforming allows casting directed beam radio waves, increasing the range and the

throughput of the transmission. gNBs deploy many antennas to support beamforming in several directions.

Another example of this greater complexity is the management of the different network slices, in which resources need to be allocated dynamically to meet the demanded QoS while keeping costs low [33]. Managing the available resources at the EC level in the MEC setting is yet another factor that enhances the complexity of 5G network management.

Cellular network operators are looking for new methods to tackle these and other challenges inherent to cellular networks, which will be accentuated in the 5G era. Research is currently being carried through the development of artificial intelligence (AI) applications, ultimately targeted at allowing service providers to increase the quality of their services [27].

Machine learning (ML) is a field of study within computer science whose goal is to create systems that learn to optimize a given problem by looking at data and extracting patterns from it [15]. ML is nowadays used in many different areas such as image recognition, recommendation systems, and natural language processing. One of the advantages of ML is that the models used to predict the different variables and hence adjust the behavior of the system can be learned automatically or semi-automatically from previous observations, requiring less direct human intervention than other approaches. This makes it a considerable approach to tackle complex problems.

As explained earlier, the 5G multi-layered network needs advanced capabilities in handling the RAN and in managing the different components, both at hardware and software levels [27]. The high complexity of the aforementioned challenges and the vast amounts of data attainable from the network elements has sparkled interest in the application of ML techniques to tackle some of the cellular network problems, as the ones explained at the beginning of this section. AI and ML applications are expected to have importance in other telecommunication areas, including customer engagement, customer care, service operations as well as network optimization [27].

2.3 Nokia's proof of concept system

Nokia Bell Labs is developing a distributed data dissemination system [23] inspired on the publish/subscribe (PUB/SUB) communication pattern. Its main goal is to provide effective communication between data sources (entities producing data) and subscribers (entities consuming data). To achieve this, the system takes advantage of data compression and efficient routing and aggregation. Nowadays, the core of the system is written in the Java programming language. The system is designed to minimize the amount of transferred data while maximizing the relative amount of up-to-date information in it [23]. To achieve this, the system adheres to the following premises:

1. No duplicate data is sent through the system. Each data point will be sent once and only once.

2. Subscribers will receive only the data they are interested in.

The 5G cellular network is a dynamic environment. From Nokia's system perspective, new data sources and subscribers will be added on the fly. Hence, the system is designed keeping in mind this changing topology.

The system was first conceived to operate in the management plane of 4G cellular networks [23]. The management plane monitors and analyzes the performance of the cellular network and its elements [47]. Key performance indicators (KPIs) are metrics calculated from performance reports of the different cellular network elements, and they are essential for this network performance management. Developments in 4G towards a flat RAN hierarchy simplified the cellular network but also removed the hierarchical processing which existed in previous generations [13]. As a result, a centralized KPI processing approach is required. Hence, Nokia's system original intent was to enable and support the processing of the high volumes of management plane data expected in cellular networks thanks to local data processing [23].

Soon afterward, many similarities were noticed between the distributed nature of both MEC and Nokia's system. Due to this, integration of the system in the MEC setting is one of today's main research areas of the project in general, and this thesis in particular. In the rest of this section, I explain the relevant details of the PUB/SUB pattern and describe the key components of Nokia's system.

2.3.1 Service architecture

Nokia's system is highly configurable due to operating a series of micro-services behind the scenes. Micro-services are pieces of software with a well-defined scope and which run independently from each other. By combining them, we can create complex behavior. Another benefit of micro-services is that it is possible to keep track of which micro-services are running on a specific EC and manage them individually. This allows us to dynamically modify the behavior of the system. We can think of a micro-service as an independent component of the system. Hence, from this point onward I will refer to them as micro-services or components.

Each micro-service hosts one or more workers. Within Nokia's system, a worker is a smaller piece of code responsible for executing some well-defined part of the micro-service's logic. For example, a micro-service might allocate a worker responsible for encrypting messages before they are sent across the network.

Nokia's system employs 2 different types of communication: PUB/SUB messaging is used for data dissemination and overall monitoring of the system, while TCP/IP communication is used for configuring the system and handling the different requests and responses involved in the messaging between the different components.

At a lower level, Nokia's system is an actor-based system. For the scope of this thesis, it is enough to know that in the actor paradigm we do not directly call methods of a worker, but we instead send messages to the worker. The worker stores all received messages in a FIFO queue and sequentially processes them. This implies that it is the responsibility of each worker to know what to do for each type of message received.

2.3.2 Publish/subscribe pattern

As mentioned in section 2.3.1, Nokia's system relies on the PUB/SUB messaging pattern for data dissemination. In the PUB/SUB messaging pattern, publishers broadcast messages that subscribers receive and consume. One way to achieve this is by assigning a topic to each message. Each subscriber then decides which topics to subscribe to, and will only receive messages from those topics. One of the benefits of this messaging pattern is that publishers and subscribers are decoupled, meaning they run without being aware of each other's existence. This allows us to modify the system architecture after the system has been deployed. In the 5G setting, this eases the management of data sources (publishers) and end-users (subscribers).

This decoupling is possible thanks to the presence of an intermediary message broker. The broker is in charge of performing the topic filtering and normally performs a store and forward function, meaning that the broker will store the message, verify its integrity and send it at a later point. There exist broker-less versions of the PUB/SUB pattern, such as the one specified in the DDS standard [60]. However, in broker-less versions, the different publishers and subscribers need to store information about each other, making them coupled. This is not desirable for Nokia's system due to having to store that information at the EC level.

In its current state, Nokia's system supports two PUB/SUB implementations, namely Apache Kafka and Apache ActiveMQ. However, any PUB/SUB messaging implementation can be integrated into the project.

2.3.3 Main components of the system

The most relevant components of Nokia's system are coordinator, data fetcher, data switch, and data hub. The concepts of stream template model and parameter are also relevant for this thesis. I describe them as follows:

- Coordinator: It is in charge of orchestrating the other components of the system. A single coordinator can handle an arbitrarily large deployment of the system. However, this is not desirable for three reasons. First, the system would have a single point of failure. Second, as the system grows, a single coordinator might not be able to handle all the requests. Third, components running on far ECs would experience a lot of delay in the communications with that single coordinator. Instead, we want to run several coordinators scattered around the network to provide a good QoS at all the targeted ECs. In multi-coordinator deployments, each component will be managed by a single coordinator, and the role of coordinator is to allow its managed components to interact with components managed by other coordinators in a transparent manner, from the component perspective. This is possible due to coordinators being able to communicate with each other over TCP/IP. Note that actual streams of data do not go through the coordinators. The role of coordinator role is purely organizational.

- Data fetcher (DF): DF's role is to ease the introduction of data from external data sources into the system. To do so, it collects data from the different sources

and performs some aggregation on it. Currently, this aggregation involves KPI calculation [23], but in the future, it could be multi-purpose. Note that data from each data source needs to be handled in a specific way, so DF must be provided with the corresponding instructions for the data sources it is serving. The aggregated data is forwarded to the next component, data switch. Note that an EC might be running various instances of DF, even though a single DF instance could suffice since each DF can handle an arbitrary number of data sources.

- Data switch (DS): DS receives the aggregated data from DF and ensures that the data is delivered to from where it was requested. Note that the same data might have been requested from several ECs. DS is composed by the PUB/SUB broker, residing outside but close to Nokia's system, and workers residing at the different endpoints of the communication. We find DS workers sending messages within DF components, and DS workers receiving messages within data hub components.

- Data hub (DH): End-users of the system make subscriptions to the system through DH. Hence, DHs act as catalogs of data sources to the end subscribers. DHs are also in charge of forwarding the requests made by the subscribers to the coordinator of their EC. Lastly, it will act as the endpoint of DS, receiving the data of the different streams and forwarding it to the end-users.

- Stream template model: Each data source requires a specific configuration for the different components and their workers. Consider a data source that produces sensitive data that needs to be encrypted before being sent over the network. Stream template models are the means through which Nokia's system retrieves the requirements of each specific data source and prepares for its handling. Stream template models contain information about which micro-services and workers need to be deployed to allow for a certain UE data stream to exist. Stream template models also provide definitions for the formulas of the different KPI values.

- Parameter: As mentioned earlier, one of DF's roles is to calculate KPI values. In more general terms, we refer to whatever value we are interested in as a parameter. A parameter is defined by its name, its data type, its data source, and its time granularity. A subscription in Nokia's system can be thought of as a collection of parameters that some subscriber wants to receive.

Nokia's system is designed with ease of deployment in mind. The configuration is loaded from external files defining the properties for every worker. Currently, research is being carried on how to automatize the deployment of the system by containerizing the different components and orchestrating the resulting containers. Said containerization is beyond the scope of this thesis.

2.4 Megasense project

In this section, I provide an overview of the Megasense project and its context.

2.4.1 Air pollution

Air pollution refers to the presence of toxic chemicals and pollutants in the air, which cause detrimental changes to the quality of life. Air pollution is estimated to be responsible for 3.1 million premature deaths worldwide every year [24].

Microscopic pollutants in the air can get past our body's immune system and penetrate in our respiratory and circulatory systems. Even though there are various types of pollutants, the type that affects people the most is particulate matter (PM). Particles with a diameter of 10 microns or less (abbreviated as PM_{10}) can penetrate deep inside the lungs, yet the ones especially dangerous are those with a diameter of 2.5 microns or less ($PM_{2.5}$). The latter can also penetrate the circulatory system, increasing the risk of heart and respiratory diseases [58].

Previous research reveals that over 80% of the world's population breathe polluted air exceeding the World Health Organization's threshold of 10 micrograms per cubic meter. Geographically, the results report that the worst levels of concentration are found on a stretch from North Africa to Eastern Asia. Figure 5, extracted from the aforementioned research, shows the first-ever long-term global map of $PM_{2.5}$. In particular, the results were obtained from combining particle measurements from two NASA satellite instruments with computer models about the vertical distribution of said particles [61].

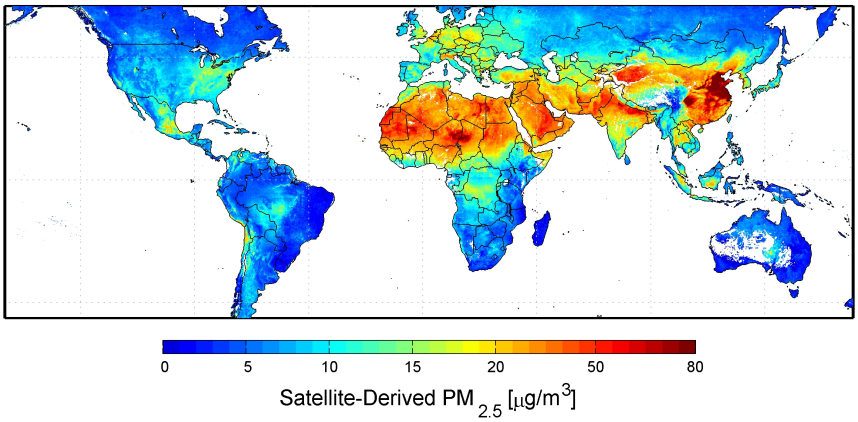

Figure 5: Global satellite-derived map of $PM_{2.5}$ averaged over 2001-2006. *Credit: Dalhousie University, Aaron van Donkelaar.*

Meteorological conditions also play a role in air pollution. On one hand, nature's air conditioning keeps the air clean. Wind mixes the gases while rain washes dust and other substances to the ground. On the other hand, strong winds can displace pollutants distances of hundreds of kilometers [59]. Hence, to get a picture of air pollution, it is not enough to occasionally measure the quality of air, but a continuous analysis of the air quality and meteorological conditions must be carried.

2.4.2 Scope and goals

The Megasense project is a research collaboration between Nokia Bell Labs and the University of Helsinki. As mentioned in section 1, the project addresses the global challenge of air pollution modeling and prediction while considering the limitations of the state of the art: low density of measurement stations and lack of high-resolution spatial-temporal data. The goal of the project is to calibrate a high number of low-quality sensors with a small number of high-quality sensors through a ML approach. These lower-quality sensors are then used to increase the spatial-temporal resolution of the system, enabling it to provide air quality analysis of large areas in a more efficient way than previous alternatives [63].

The first pilot was launched in 2017 in the city of Helsinki [57]. The next pilot will be launched in Beijing, China [56]. Taking Beijing as an example, research from 2006, when the city population was nearly 11 million people, categorized the city as having serious PM pollution issues and assessed the acute effects of air pollution regarding several health aspects, including mortality [25]. Nowadays, with almost 22 million population [62], we can only expect the situation to have worsened.

Hence, the ambition of the Megasense project is to develop and enhance new ways to collect more and better air quality measurements. This will ultimately allow governments and other parties to improve people's quality of life.

The project considers various types of sensors for data collection, namely high-end scientific instruments, commercial air quality transmitters, dense low-cost sensor arrays and consumer wearables [57]. Table 2 provides a comparison between two particular sensors utilized in the project, the high-end stations measuring earth surfaces and atmosphere relations (SMEAR) and the Vaisala AQT410 commercial sensor [53].

Sensor name	Mobility type	Sampling period	Pollutants measured	Cost (euros)	Units deployed (Finland)
SMEAR	Stationary	1 s	> 1200	~1 million	4
Vaisala AQT410	Portable	10 min	4	~3.3 K	TBD

Table 2: Comparison between two types of sensors used in the Megasense project [53] [54] [55].

Nokia's system is Megasense's backbone regarding data dissemination. Currently, sensors generate data that is then analyzed in a centralized server on the cloud. This approach will run into performance issues as the number of sensors and collected data points increases. The problem will be further aggravated by the inclusion of sensors attached to moving entities such as buses, cars, bikes or even clothing apparel.

Hence, Nokia's system must allow for the carrying of the computations close to the sensors, as the sensors traverse the grid. As mentioned in section 2.2.4, this achieves low latency, enabling for better real-time analysis, and reduced network bandwidth usage, improving the system in terms of scalability.

3 Data processing handover and solutions

In this section, I provide a thorough description of the work done for this thesis. First, I provide an introduction to the HO problem. Then, I provide further context by relating it to 5G and self-driving cars. I conclude the chapter by explaining the different HO procedures designed for Nokia's system.

3.1 Rationale

In this section, I compile the most relevant considerations that the reader must be aware of regarding the HO problem. I also outline the most relevant characteristics that a solution to the problem must achieve, as in the case of Nokia's system.

3.1.1 The handover problem

In section 2.2.4 I provided an overview of the MEC paradigm. For it to be usable in real-life scenarios without jeopardizing QoS, the HO problem must be addressed.

For this thesis, I define the HO problem as the set of technologies and solutions that will enable UE in the MEC setting to traverse the grid without impacting the system's performance. The HO problem can be divided into three subproblems: user HO, service HO and data processing HO.

- User HO: Consider some particular UE. As it traverses the grid, the gNB emitting the strongest perceived signal will vary. The user HO problem refers to the challenge of keeping the UE connected through the optimal signal at all times to maximize user QoS.
- Service HO: As explained in section 2.2.4, in the MEC setting ECs are supplied with small data centers. The network and the services need to be optimized so that said data centers can launch new services on demand with minimum overhead. The service HO problem refers to this challenge. The service HO problem is related to the concept of service migration in cloud computing.
- Data processing HO: It might be the case that a specific service requires some external data to operate. In particular, for the Megasense project, DFs will require different KPI formulas to support different data sources. The data processing HO problem refers to the challenge of fetching and providing these formulas/data on demand with minimum overhead.

Relating the HO problem to the Megasense project, an issue that arises with the inclusion of mobile sensors is how to guarantee that the subscribers will receive a consistent stream of data, regardless of where the sensors are and move to. Since one of the goals of Megasense is to attach sensors to mobile agents such as buses or even pedestrian clothing, it is important that the HO problem is issued within Nokia's system. Hence, this is where the main research of this thesis will be focused on.

The work presented in this thesis relates the most to the data processing HO problem. In particular, I want to emulate the data processing HO in Nokia's system,

right after the user HO has taken place. That is, after some UE has established a connection to a new EC. This allows me to abstract the implementation away from low-level 5G specification details regarding the HO procedure. In particular, details regarding how the UE and the gNB communicate are omitted. The work presented in this thesis assumes that the network has already enabled the user HO.

Furthermore, I also assume that Nokia's system is deployed close to all relevant ECs, enabling the system to attain the desired QoS without worrying about system deployment. This means that the service HO logic is out of the scope of this thesis, making all work presented tightly related to Nokia's system and its actual implementation.

3.1.2 Problem definition

In section 3.1.1 I described the HO problem and distinguished three related subproblems. For the work presented in this thesis, emphasis is put on how to guarantee that end-users receive a consistent stream of data, regardless of where the data sources, to which the user is subscribed, are and move to. Hence, the subproblem that relates the most to the work of this thesis is the data processing HO one.

The ambition of the Megasense project is to provide more and better insights about air quality in specific areas thanks to the inclusion of data gathered from mobile sensors. That way, they can geographically cover areas that would be otherwise unreasonable to include for different reasons, but mainly high deployment and maintenance costs for stationary sensors. An appropriate solution for the HO problem must ensure that:

- Data is delivered to where it is needed, and only to where it is needed. This might seem obvious, but cannot be overstated. Since Nokia's system is thought to be scalable, to optimize resource usage, it is important that no component in the system receives unrequested data. This aspect is also relevant concerning data privacy and the security of the system. Research is ongoing regarding those aspects by the time of writing this thesis.

- Data is delivered promptly. As shown in Table 1, different cellular network generations have different latency requirements. Relating to Megasense and Nokia's system, achieving minimum delay is a priority if air quality monitoring is to be used for fire detection or gas leakage detection, for example.

- Data processing HO takes place without any data losses. For critical cases, such as self-driving cars, we cannot accept any data losses. However, a flexible solution should allow the user to decide whether losing some data in exchange for efficiency is acceptable.

- Resource usage is optimized. Optimization plays a huge role in the MEC setting since resources at the EC are limited and shared across possibly many different services. Optimization can be the difference between a theoretical solution and a solution that can be deployed in real-life scenarios.

Therefore, the problem addressed in this thesis revolves around the design and implementation of a solution that aims at allowing data processing HO within Nokia's system. The data processing HO must be transparent from the subscriber perspective and the solution must address all of the four aforementioned conditions.

Other, perhaps simpler, related problems are how to act in case of the gNB-UE connection being lost and how to optimize sensor battery usage by closing the connection when no data is available to be sent.

3.2 State of the art

As described in section 2.1, in cellular networks the last link between UE and the BS is wireless. Hence, the HO problem is inherent to cellular networks since the invention of portable UE. Before 5G, technical developments allowed only for break-before-make solutions. This means that in the case of connecting to a different BS, the connection to the source BS is first closed (break) and only then the connection to the target BS is established (make). This results in the connection being inevitable closed, even if just temporarily.

In this section, I provide an overview of the current state of the HO problem, with emphasis on the 5G cellular network. I also discuss some of the most relevant concepts which involve HO in the context of self-driving cars, due to the many similarities in common with the Megasense use case.

3.2.1 Handover and 5G

Previous research evidences that 4G LTE specifications will not reach the QoS requirements for new services and use cases that 5G will enable [8]. The authors report a median user HO time of 40 ms and radio link failures (RLF) occurring in approximately one percent of the HOs. The 40 ms median time is too long for critical situations, such as avoiding a collision in the case of self-driving cars. The one percent link failure is also unacceptable due to longer delays derived from having to re-establish the connection. The authors conclude that the 5G design must utilize new mobility methods such as make-before-break connectivity, multi-cell connectivity, and synchronized HOs to truly meet the 5G specification requirements.

As opposed to break-before-make connectivity, make-before-break allows for the connection to the new BS to be established before closing the original connection. This allows for HOs with zero downtime and no data loss. Make-before-break is possible thanks to multi-cell connectivity, which means that the 5G specification considers for UE to simultaneously establish connections to different gNBs. Note that cellular network generations previous to 5G only allowed for single-cell connectivity.

Break-before-make HO mechanisms were designed to work under asynchronous networks, meaning that the clocks at the different cells were not synchronized. Synchronized HO refers to enhancements in the HO procedure related to the recent trend of having time-synchronized cells [2].

The last step of a break-before-make HO requires the UE to perform a random access (RA) in the target cell to acquire its timing advance. This process results in

an extra delay for the HO. Let me explain it more in-depth, taking 4G LTE as an example [65]. Note that the 4G cellular network needs to support a large number of UE. To achieve this, a single radio frequency is shared by several UE. More precisely, each radio frequency is divided into eight timeslots, corresponding to the maximum number of UE which that frequency can handle. Then, the eNB will synchronize the UE so that they only transmit information during their allocated timeslot. For completing the HO, the UE must establish a connection to the target eNB. To do so, the UE will randomly pick an available preamble sequence from the target eNB and try to connect through it. This is the RA to the target cell. At this point, there might be a collision due to other UE also trying to establish a connection over the same preamble sequence. In that case, only one of them will succeed and the other will need to retry the connection after timing out. Upon success of the connection, the eNB estimates the transmission timing of the UE and sends back a response that contains the timing advance command. The eNB considers the estimated transmission delay and sends the response only when it expects the UE to receive it in a synchronized manner. That way, the UE knows which of the timeslots to use for uplink transmission. Further details of this process are out of the scope of this thesis.

In contrast, in time-synchronized networks, the RA to the target cell step can be avoided. In [2] they propose a solution that relies on time-synchronized networks to decrease HO overhead. In particular, their solution relies on the source and target cells arranging a time at which the former will close its connection and forward the upcoming data to the latter, while at the same time the latter will start allocating resources to the UE for its uplink transmission. After that, once the UE sends confirmation to the target cell about the HO, normal transmission can be resumed. The authors report a reduction in HO time from 55 ms to 5 ms with that approach.

The 3GPP is in charge of developing and updating the 5G standard through different specifications. The standard is updated through different so-called releases. 4G LTE release 12 (March 2015) introduced several enhancements regarding mobility, such as mechanisms for reducing the probability of having high-velocity UE connecting to cells, faster recovery from RLFs, context fetching after RLFs and UE transmitting their mobility history to the network to allow for better management [2].

Another radio technology standard worth mentioning is the narrowband internet of things (NB-IoT) standard, also developed by the 3GPP [31]. NB-IoT is optimized for automated data communications between devices, also called machine type traffic.

NB-IoT considers communications based on short and infrequent messages, and one of its goals is to extend the battery life of the IoT devices by keeping UE connections closed for most of the time [31]. Due to the short nature of the messages, NB-IoT assumes that a message can be sent if UE is within network coverage. This means that the standard does not consider cell HOs since the UE will send all the messages it needs to send and then close the connection. The UE might only connect to a new cell when establishing a connection from idle mode. The NB-IoT standard specifies that BSs continuously emit a specific signal, whose purpose is to let the different UE analyze the quality of the connection, regardless of when the UE has active connections. When a UE wakes up from idle mode, it will measure the received

power and quality of the aforementioned signal sent by every BS. If the UE finds a signal with power and quality levels above some defined thresholds, it will camp on the cell of that signal. However, if the power of the signal degrades below a certain threshold, the UE might need to start a cell re-selection process on the next connection. If that is the case, it will simply check the quality and power of all incoming signals, and then rank them. Then, the UE will select the highest-ranked cell which is suitable, that is, that it can provide normal service, and connect to it.

As mentioned in section 2.2, the 5G and MEC setting will enable new business opportunities. However, it also demands new technologies and solutions to the HO problem, since now not only UE is traversing the grid, but also there is a need to keep the services close to the UE to provide a good QoS. The main challenges are how to perform the service migration, how to handle resource allocation and how to monitor and configure the systems in such a dynamic environment.

In the current literature, we find interesting concepts and ideas related to service HO, such as intelligent containers [14]. The authors propose a system based on a hierarchical deployment of the services. The idea is that at the time of deployment, the original instance of the service, to which the service provider has access, is the root of some tree. The service will then automatically replicate and migrate to different servers, based on incoming request patterns. For example, suppose the system starts receiving requests from some far-away area. To improve the QoS in that area, the system might decide to replicate or migrate to another server closer to that area. In the case of replication, the new node will be considered a child of the node it replicated from, making the tree grow organically. The hierarchical nature of the tree serves two purposes. First, it allows for monitoring of the system with minimal overhead, since each node can monitor itself and then forward only relevant information to its parent. That way, the system can still be globally monitored from the root perspective without the need to send all data to it. Second, it allows the service provider to manage the resource allocation of the different instances. The service provider can set rules to automatically or semi-automatically manage the deployment across different nodes and assign a budget limit. The system will then automatically adjust itself, trying to maximize its performance under those restrictions. Said performance can be measured in various ways. The authors propose an evaluation metric that takes into account the number of requests per second experienced and the latency experienced by users from a given subnet.

3.2.2 Handover and self-driving cars

Traditional HO mechanisms are based on a reactive approach, in which the UE reacts to signaling indicating changes in the network connectivity as the UE moves around. In contrast, proactive HO approaches, in which the UE actively decides when and where to HO, can result in more efficient and reliable HO mechanisms. The goals of proactive HOs are to minimize both packet loss and service disruption times [3]. There also exist other approaches to mitigate the impact of HOs in the context of self-driving cars.

Self-driving cars are the result of advances in various areas related to the auto-

motive sector, namely autonomous vehicles and vehicle-to-everything (V2X) communication.

Autonomous vehicles are vehicles equipped with a subsystem of onboard sensors. Through those sensors, the vehicles can build a map of its surroundings. However, relying on that alone, different vehicles are not able to cooperate, which prevents them from performing complex maneuvers efficiently, limiting their potential. Hence, it is crucial to provide vehicles with means to share and aggregate the data they gather. The aggregated data can then be used for two main purposes: cooperative sensing and cooperative maneuvering. Cooperative sensing allows several nearby vehicles to construct a better understanding of their surroundings, while cooperative maneuvering allows for better orchestration of a fleet of vehicles. Together, they increase traffic safety, efficiency and comfort [12].

V2X refers to communications between vehicles and any other identities. V2X encompasses other more concrete types of communications, such as vehicle-to-infrastructure (V2I), vehicle-to-vehicle (V2V) and vehicle-to-pedestrian (V2P).

For the case of the self-driving car, radio resource management (RRM) algorithms can be designed to take into account the density of the road according to different times. For example, we can expect a higher load on a certain road during rush hours. Trajectories of the vehicles can also be predicted with information about their path, enabling for smoother HOs [4].

How to support and guarantee V2X communication is a research area that attracts interest from governments, companies and academia [16]. A lot of effort has been made to enable V2X through both already existing technologies and newly developed ones. Let us take a look at the most relevant technologies and their relation to V2X communications.

Older types of communication such as WiFi and infrared technologies have been reported as not suitable for high mobility scenarios, mainly due to low reliability, unbounded delays, and intermittent V2I connectivity. 4G LTE networks are suitable for handling V2I communications, due to their high data rate, high penetration rate, and large coverage specifications. However, 4G LTE networks struggle when applied in V2V communications due to their centralized architecture and the heavy load generated by periodic messages [4].

At this point, it is worth discussing vehicular ad-hoc networks (VANETs). VANETs are wireless networks created and supported by the vehicles themselves. Ad-hoc networks are cooperation based networks, meaning that the network topology is decided and influenced by the network elements themselves [17]. As a consequence, the topology of a VANET will be in constant change. The main goal of VANETs is to enable V2V communication without the need for cellular coverage. However, the HO problem is also present when using VANETs due to VANET network characteristics, namely high velocity of the components, small coverage range and different mobility patterns [3]. In [18] they explore different ways in which the HO can be optimized in the context of VANETs. The idea is to take assistance from other vehicles during a HO, to reduce HO latency and minimize packet loss. This can be done by a vehicle getting information from another vehicle about the next access point before the HO takes place, resulting in considerable HO reduction times.

One of the first efforts in the direction of supporting proper V2V communication through VANETs is the so-called IEEE 802.11p. 802.11p is an approved amendment to the IEEE 802.11 standard to add wireless access in vehicular environments. Despite lots of research being carried about how to develop V2X communications using 802.11p, 802.11p has been proven to face reliability and scalability problems as the network load increases [19]. For this reason, 3GPP included support and specifications for V2X communications around 4G LTE in release 14 (September 2016). Specifications of this release are typically referred to as LTE-V (for vehicles).

Now let us focus on how LTE-V addresses V2V communication, since V2I communication is tractable through previous LTE releases. LTE-V considers four different communication modes for V2V. Mode-1 and mode-2 are known as device-to-device (D2D) communications. They were first introduced in release 12 and define a communication interface known as sidelink or PC5 interface. This interface enables UE to communicate directly to other UE without the need for a connection to the eNB [20]. However, these two modes were designed to increase UE battery lifetime, which results in increased latency for the communications. As a consequence, mode-1 and mode-2 are not suitable for V2X communications [19].

The other two modes, namely mode-3 and mode-4 were introduced in release 14. They differ in how the radio resources are allocated. In mode-3, the resources are allocated by the cellular network, while for mode-4 the resources are allocated by the vehicles using a distributed scheduling scheme [19]. Hence, mode-4 can operate without cellular coverage. This makes mode-4 an alternative to 802.11p. Particular details of how mode-4 operates are out of the scope of this thesis.

Regarding the types of messages in V2X, the European Telecommunications Standards Institute (ETSI) classifies messages into two types: cooperative awareness messages (CAMs) and decentralized environmental notification messages (DENMs). CAMs are periodic messages used to maintain a shared knowledge between the different endpoints or stations on some ITS network, supporting the cooperative performance of the vehicles present in the road network [21]. DENMs are used to alert about abnormal or dangerous traffic conditions. DENMs are disseminated to other intelligent transport system (ITS) stations either through V2V or V2I communications [70].

Researchers compared the feasibility of 802.11p and 4G LTE for V2X communications on a real test-bed which included 802.11p roadside units, 4G LTE cellular communication stations, and vehicular onboard terminals. Results indicate that 4G LTE is more suitable for applications that do not require high speed and reliability for communications, such as traffic information transmission or downloading a file. However, for critical applications, such as collision avoidance, 802.11p outperforms 4G LTE [22].

Researchers have also shown that LTE-V outperforms 802.11p when the latter is configured with the default data rate of 6 Mbps. However, different optimizations can be applied to each method to improve its performance, namely increasing the data rate for 802.11p and increasing the modulation and coding scheme for LTE-V [19]. These results indicate that LTE-V will set the baseline for V2X communications in the years to come.

3.3 Methodology

Before any work of this thesis was integrated into Nokia's system, it was possible to manually emulate a HO to some extent. However, the lack of some features made it impossible to use those results as a baseline for comparison. The main drawbacks encountered were:

- Unreliable and tedious process. Since the user would have to manually shut down and start the different data streams, results would be hardly reliable. As an example, it would be difficult to accurately measure the total time that passes between a stream is shut down and then restarted on the target EC, due to added delays of the user having to navigate through the user interface.
- The topic of the PUB/SUB messaging would not be reused, which would make it even more undesirable for real-life deployment. Keep in mind that in the MEC setting, it is important to try to reuse components and keep resource usage to a minimum. Not reusing the PUB/SUB topic implies needing to send additional messages to the different components for the new configuration, with its associated overhead.

For those reasons, I needed something more reliable that I could use as a baseline to compare the results of my final solution. As a result, I designed two HO procedures, which I named *break-before-make (BbM) HO* and *make-before-break (MbB) HO*. The main difference between them comes from the cellular network generation they are inspired on. For the *BbM HO*, I wanted to emulate technological limitations present in older cellular networks, specifically the break-before-make limitation to connectivity. As for the *MbB HO*, I wanted it to take advantage of the make-before-break approach that 5G will enable. As we will see, this has the benefit that it will enable for seamless HO, while at the same time requiring a more complex solution, especially if we want to use it in cases in which no data loss is affordable (more about this will be discussed in section 5.1).

3.3.1 Break-before-make handover

The *BbM HO* emulates a simpler version and logic of execution for the data processing HO within Nokia's system. The overall idea is that upon detection of a data source moving towards a new EC, we first break the UE connection to the source EC and only then create a connection to the target EC.

This is the case of, for example, users roaming around with their smartphones while connected to the 4G network. Note that from the human perspective, the time in which the connection is interrupted is barely noticeable. Previous research reveals HO execution times of 40ms at the median level for 4G LTE networks [8].

For Nokia's system and in the case of a HO, this could be more noticeable due to having to wait for the initialization of DF in the target EC - assuming that DF was not already deployed. This will be further discussed in chapter 4.

Let me illustrate the logic of the *BbM HO* with an example. Figure 6 shows the starting configuration of the example at hand. As we can see, we have three

different ECs, namely Edge1, Edge2 and Edge3, each of them running an instance of coordinator, to provide access to Nokia's system functionality within each EC.

Also, let us consider that a stream is running from a data source through DF up to DH. In Figure 6 these components are labeled as Data Source, DF1 and DH, respectively. This example belongs to the multi-EC scenario, in which DF1 and DH are present in different ECs.

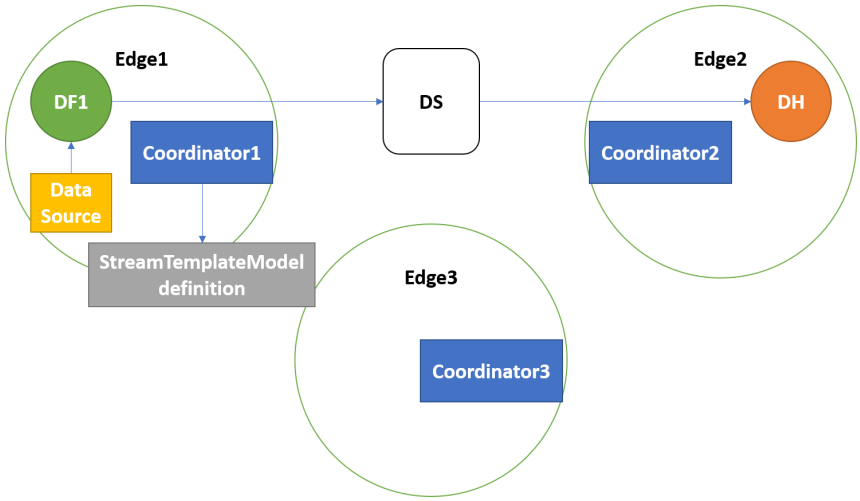

Figure 6: Configuration of Nokia's system before any HO.

At some point, Data Source starts moving towards Edge3, finally establishing a connection to it, as depicted in Figure 7.

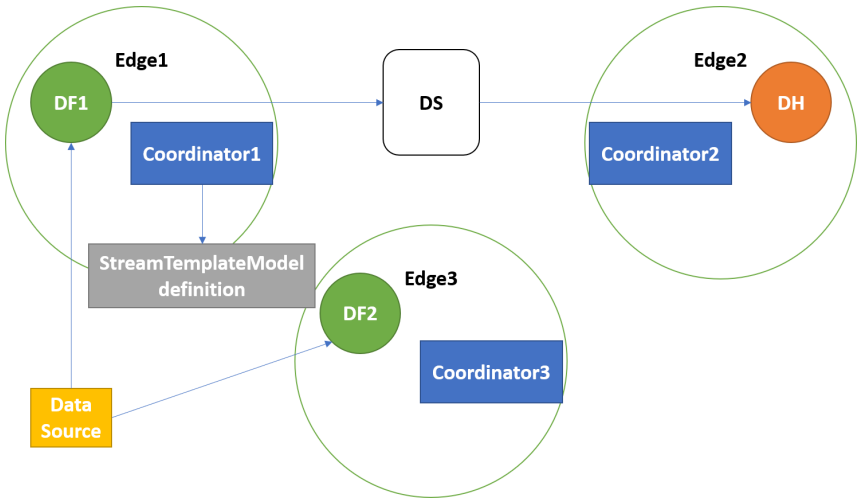

Figure 7: Data Source changes its position, establishing a connection with the target EC.

When this happens, Coordinator3 spawns DF2 notifies all the other coordinators of the system about Data Source now being available from DF2. Upon reception of that message, the other coordinators will forget about Data Source being available

from DF1. Also, since Coordinator3 had information about in which EC Data Source was before the HO, it will send to the coordinator of said EC, Coordinator1, a request for retrieving information about the streams involving Data Source. This is depicted in Figure 8.

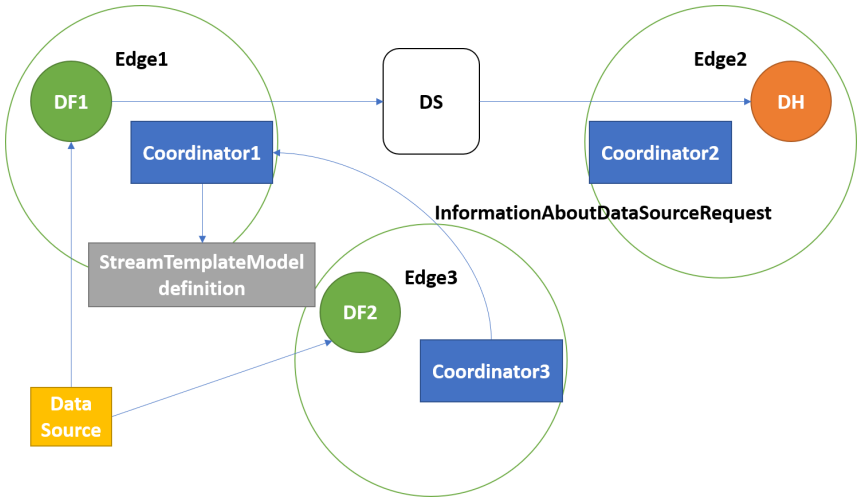

Figure 8: The coordinator of the target EC sends a request to the coordinator of the EC to which Data Source was originally connected to.

When Coordinator1 receives the message, it will first shut down the streams involving Data Source. This is depicted in Figure 9. How this is exactly done behind the scenes is out of the scope of this thesis.

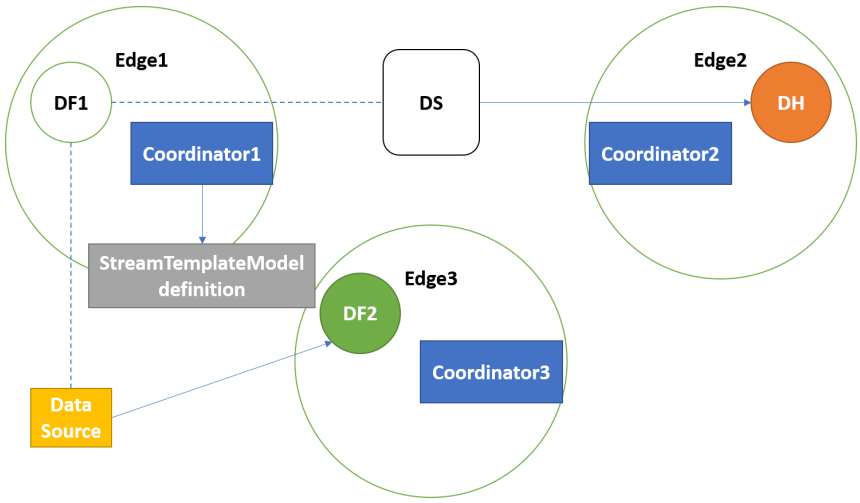

Figure 9: Coordinator of the source EC shuts down the streams of the parameters related to Data Source.

Upon confirmation of the stream being successfully shut down, Coordinator1 will send a response back to Coordinator3. This is depicted in Figure 10. Within that response, there is information about the different streams and topics related to Data

Source and information about the stream templates needed by the target EC DF to set up the stream. Recall that the stream templates are what lets DF know how to translate the raw data points into useful KPI values.

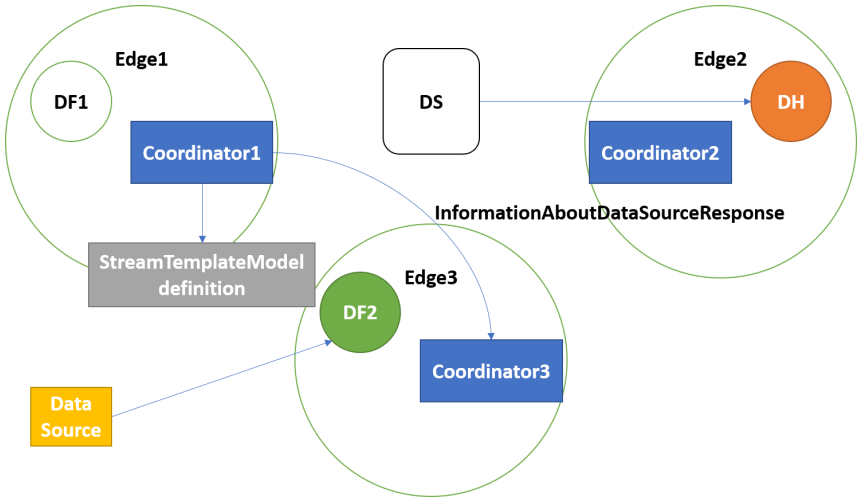

Figure 10: The coordinator of the source EC sends a response back to the coordinator of the target EC.

Once Coordinator3 receives the response, it is finally able to forward the relevant information to DF2, and so the same stream will be reused upon new data generated by Data Source. This is depicted in Figure 11.

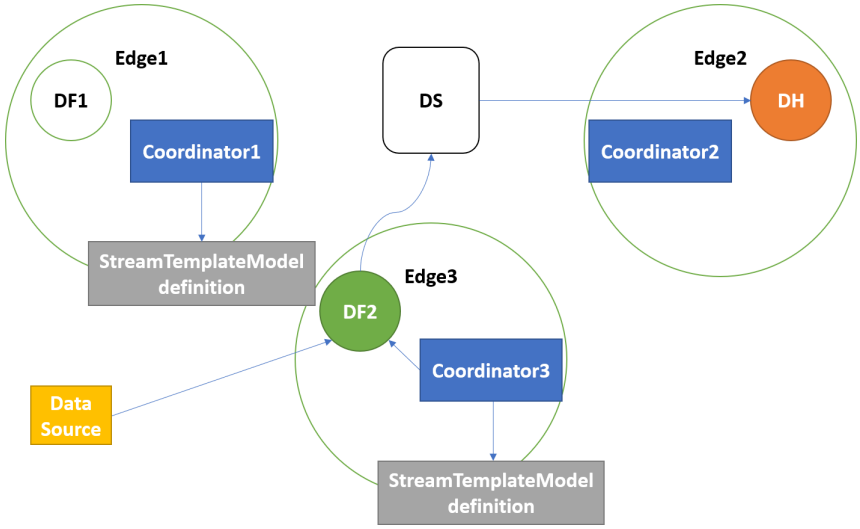

Figure 11: The coordinator of the target EC supplies the information to its DF, enabling the stream to be recreated.

As I will discuss in the results chapter, I consider the HO to have finished successfully once the DF of the target EC is done setting up the stream components.

Note that this solution depicts a break-before-make approach. This means that from the perspective of DH, the stream will not be stopped but there will be a

noticeable pause on it if the time granularity selected is low enough, due to delays in the time it takes to set up a stream by DF. In consequence, data might be lost when using this approach.

Figure 12 depicts the configuration of the system after the HO is complete. As we can see, now the data stream goes across Edge3 instead of Edge1, as intended.

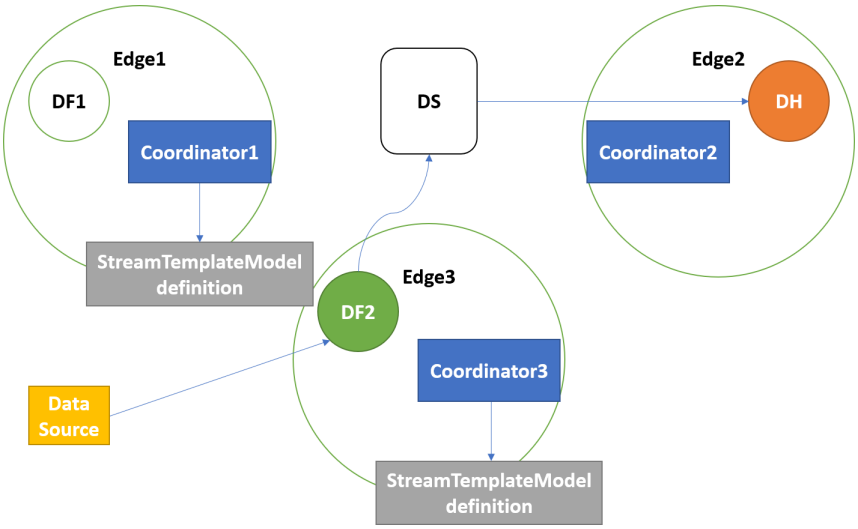

Figure 12: Once the stream is only active from the target EC, the HO is complete.

3.3.2 Make-before-break handover

As we saw in section 2.2.2, 5G demands the capability of executing a zero HO execution time. This will be possible thanks to 5G supporting several connections from the UE to different edges simultaneously, also known as multi-connectivity [26]. In other words, the UE can connect to the target EC while being still connected to the source EC. This effectively enables for make-before-break HOs, since now we are first establishing a new connection (make) before closing the old one (break), so there is always a path for the data to traverse the network.

Let me explain this HO solution with another example. The starting configuration I consider is the same than for the *BbM HO*, explained in section 3.3.1 and pictured in Figure 6. Let us recall that the configuration represented a multi-EC scenario.

At some point, Data Source will start moving towards another EC. When this happens, Coordinator3 will serve it by spawning DF2, as depicted in Figure 7.

Again, Coordinator3 knows where Data Source was previously, so it will send a request to the coordinator serving it on its former EC, Coordinator1 in this example. This is pictured in Figure 8.

At this point, the two proposed HO procedures diverge. Instead of first shutting down the stream, Coordinator1 will directly respond to the request with the same information than in the old HO case. This is pictured in Figure 13. As we will see in the results in section 4.3.1, this makes the response to happen faster due to avoiding the need of having to wait for the stream to shut down at the source EC.

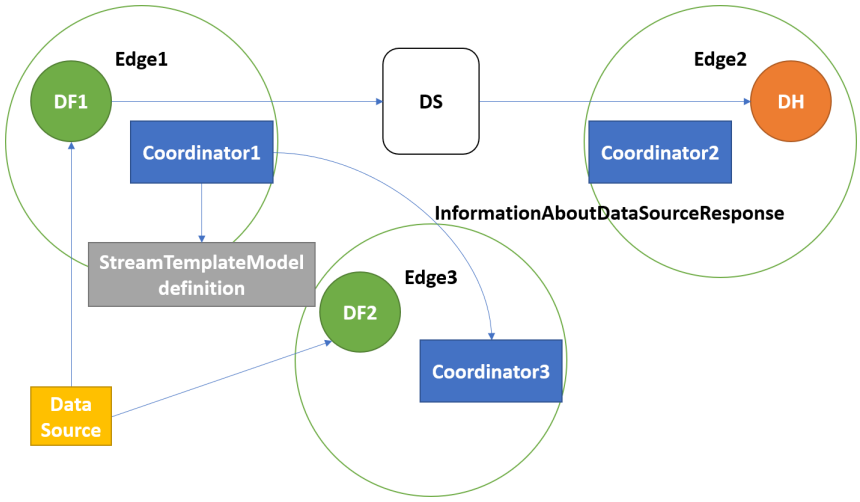

Figure 13: The coordinator of the source EC immediately sends a response to the coordinator of the target EC.

Coordinator3 will receive the response with the information about the stream topics and the stream template model and will forward it to DF2 so that the stream can be started from Edge3. This is pictured in Figure 14.

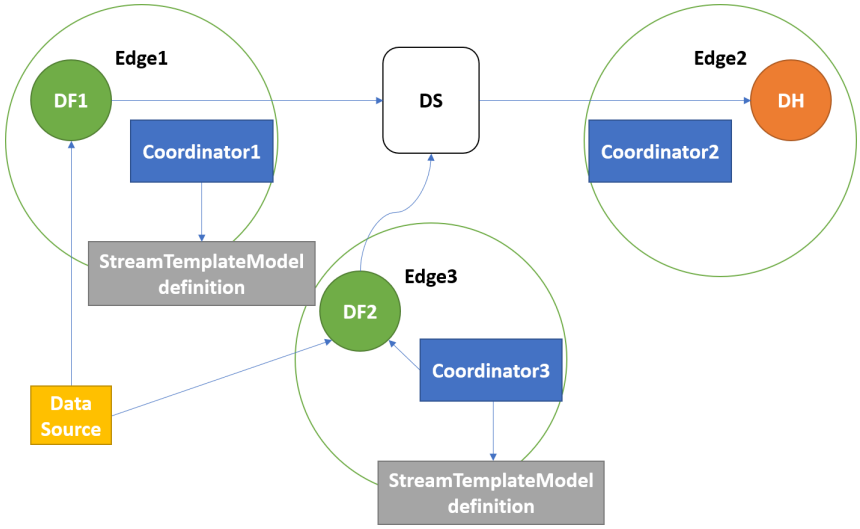

Figure 14: The coordinator of the target EC supplies the information to DF2, enabling the stream to be recreated.

Once the stream is set up in Edge3, Coordinator3 will send Coordinator1 a message to inform it that the stream has been set up on the target EC, and hence Coordinator1 can shut down its part of the stream to avoid sending duplicated data and minimize resource usage. This is pictured in Figure 15.

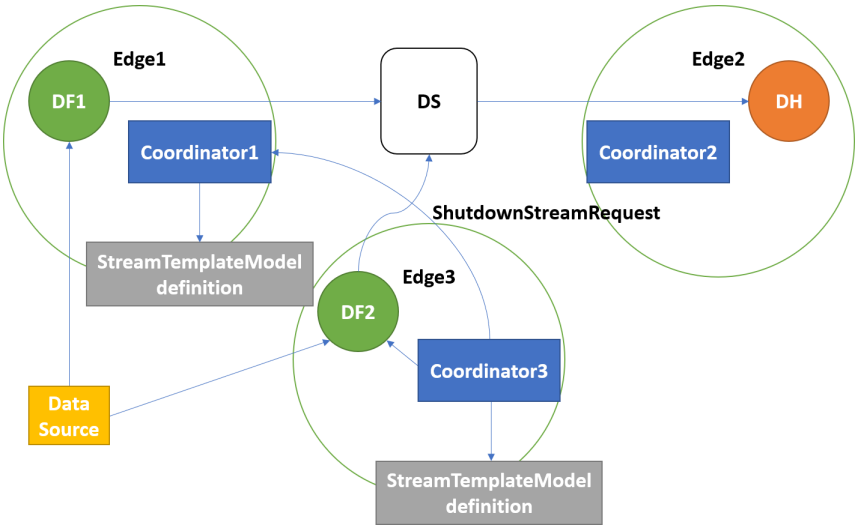

Figure 15: Once the stream has been created in the target EC, the coordinator of the target EC will inform the coordinator of the source EC that it can now shut down its part of the stream.

From the perspective of DH, no data is lost with this procedure but duplicated data might be received, coming from both ECs, in the time it takes for Coordinator3 to tell Coordinator1 to shut down its part of the stream. Solving this stream data duplication is not a trivial problem. I discuss this problem further in section 5.1. The end configuration of the system is the one pictured in Figure 12.

3.4 Enhanced make-before-break handover

In section 3.3 I explain the two HO procedures integrated into Nokia's system as part of the work for this thesis. These procedures serve to show the usefulness and suitability of the *MbB HO* procedure. However, before deploying the system on real scenarios, it would be beneficial to enhance it by taking into consideration some factors related to the 5G network.

Hence, Figure 16 depicts my proposal for the final solution. I propose the creation of a new component, the user equipment interface (UEI) agent, which is in charge of handling the requests from data sources coming from other ECs. The data sources know about its presence thanks to self-discovery algorithms at the target EC. The UE agent will simply forward those requests to the coordinator of the target EC. Coordinator will make sure a DF is ready to serve the HO, spawning it if not already present. On confirmation of DF being ready, coordinator of the target EC will send the information about said DF to its DF, who will then forward it to the data source. With that information, the data source can now establish a connection with DF from the target EC. DF at the target EC will set up the necessary streams. Keep in mind that this requires DF to fetch the stream template models. Whether they are fetched from the target EC DF or somewhere else is out of the scope of this thesis. Once the streams are set at the target EC, DF at the target EC notifies its coordinator about it. Coordinator will then forward the notification to the coordinator of the target

EC to allow for the graceful closing of the stream. Coordinator of the target EC will then tell its DF to close the streams, freeing the resources and completing the HO.

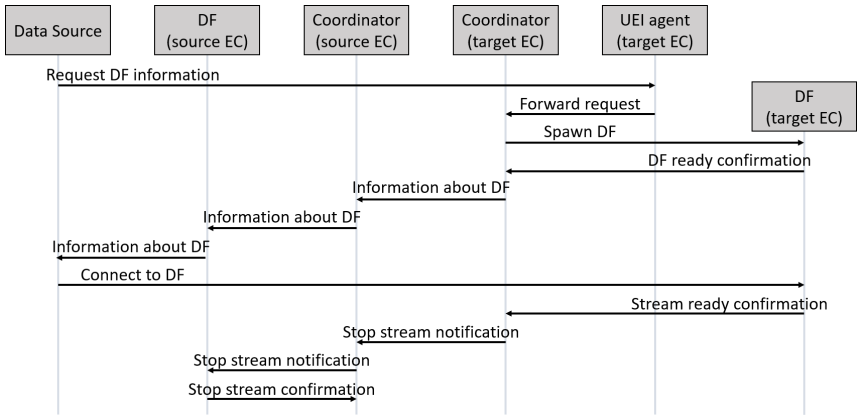

Figure 16: Proposed workflow for further revisions of the *MbB HO* procedure.

As we can see, the main innovation of this proposal is the UEI agent and its self-discovery process. However, further details are out of the scope of this thesis.

3.5 Implementation

In this section, I list the most relevant contributions of this thesis into Nokia's actual system.

- Implementation of the communication scheme between coordinators on data source HO. In particular, when a coordinator is notified by DF about some data source, and the coordinator already knew about said data source being present in some other EC, it will update its information and notify other coordinators about it. Hence, it is the coordinator of the target EC who starts the HO within Nokia's system.
- Make coordinators store which data sources are involved in a HO until the HO is complete. This allows the coordinator to close the streams at a later point, even if the data sources are marked as being present in a different EC by the time of closing them.
- Improve logic of coordinator components, allowing them to skip generating the stream instructions for DH if a HO is taking place, minimizing resource usage.
- Make DF and DH notify their coordinator upon finishing their stream related tasks. This allows for more fine-grained control of the system. In particular, this feature is needed to guarantee that the *BbM HO* solution is truly BbM, by waiting for the original stream to close in the source EC before sending the required information to the target EC.
- Make coordinator shut down the streams one at a time. Otherwise, the system crashed due to a bug, since HOs were not considered.

4 Evaluation

In this section, I explain the most relevant results obtained and their practical implications. All the results and measurements reported in this thesis have been gathered directly from the execution logs of Nokia's system.

I opted to emulate each HO solution three times, for a total of six executions. To minimize the impact of external factors, the IntelliJ [45] integrated development environment (IDE), used for the implementation of the solution, and the Kafka [46] broker were the only user programs running in the operating system during the measurements. I also restarted the system before each execution to avoid the possibility of the program being stored in memory for subsequent executions.

For each of the six HO executions, I opted to emulate the HO of a single UE between two ECs. The rationale behind this decision is to be able to showcase the different implementations and their theoretical implications while avoiding any 5G network emulation procedures, which is a far from trivial problem beyond the scope of this thesis. I discuss this later in this chapter.

On a lower level, I employed the same UE configuration for all the HO executions. The emulated UE gives access to a total of 165 parameters stemming from 11 individual sensors. These are emulated locally, reading the data from a file stored on disk. The time granularity selected for all the subscriptions was 2 seconds.

4.1 Testing environment

Table 3 provides an overview of the configuration used for the evaluation of the solutions implemented in this thesis.

Laptop	Lenovo Thinkpad T480
Processor	Intel Core i5-8350U @ 1.7 GHz
Cores	4
Logical processors	8
L1 cache	256 KB
L2 cache	1,0 MB
L3 cache	6,0 MB
Memory	8 GB
Operating system	Windows 10 Enterprise

Table 3: Testing environment configuration.

No virtual machine or further special configuration was used for the evaluation of the solutions. This results in the following benefits:

- No clock-related issues thanks to running everything on the same machine and operating system.
- Closest to theoretical comparison possible.
- Easier configuration and replication of the results.

4.2 Overhead of spawning data fetcher

Before diving into the actual results, I would like to speak about a factor that plays a role in the measurements of this thesis. In particular, when emulating the HO procedures, I assume that no DF is present at the target EC. Hence, the way I am emulating the HO involves the instantiation of DF in the target EC, right before starting the HO.

In the current implementation of Nokia's system, said DF is required to trigger the HO mechanism since otherwise the data source has no way to communicate with the coordinator. In section 3.4 I proposed some enhancements to the system to solve this issue, yet further details are out of the scope of this thesis. A single DF can handle an arbitrarily large number of data sources, so running a single DF at the target EC is enough.

The overhead of instantiating a DF in the environment described in section 4.1 was about 10 seconds. This time was not taken into consideration for the results discussed in the rest of this chapter. There are two reasons for this decision. First, on a conceptual level, it does not make sense to include said time in the calculations. Since it is a one-time delay per HO, the average will be biased, especially when emulating a HO for a small number of data sources, as is the case of this thesis with only 11. Second, the whole overhead can be avoided in practice by running a DF instance ahead of time and then emulating the whole HO.

4.3 Results

Let us dive now into the actual results and their interpretation. In this section, I discuss the feasibility of the different solutions while considering 5G related specifications.

4.3.1 Stream managing overhead

In this section, I go over the different measured times related to the execution of the different HO solutions. I then provide an overview of how well each solution would perform in the 5G and MEC setting when considering different mobile entity speeds, such as bicycles, cars, and trains.

First, let us take a look at the results of the *BbM HO* executions, summarized in Table 4. As we can see, the average time for stopping the stream at the source EC is 140 ms and the average time for restarting the stream at the target EC is 78 ms. From the perspective of the target EC coordinator, the average time it takes to receive a response from the source EC coordinator as depicted in Figure 10 is 140 ms. As we can see, the times for getting a response match those of stopping the stream at the source EC. This is because, in the *BbM HO*, the coordinator of the source EC waits for confirmation of the streams being shut down before sending the response. The values in parenthesis indicate the time after which the response was ready to be sent. However, since sending the response without confirmation of the streams being shut down does not guarantee a true break-before-make solution, it was left out of the implementation for this thesis.

	BbM HO 1	*BbM HO* 2	*BbM HO* 3	Average
Avg. stream stop time	91 ms	139 ms	190 ms	140 ms
Avg. stream start time	57 ms	81 ms	97 ms	78 ms
Avg. request response time	91 (48) ms	139 (102) ms	190 (66) ms	140 (72) ms

Table 4: Summary of the observed times regarding the handling of the streams for the *BbM HO* executions.

We can derive the average time for a *BbM HO* to be 218 ms. This is the result of adding the average times of getting a response to the request (140 ms) and starting (78 ms) the streams. The main events of the *BbM HO* are illustrated in Figure 17. As we can see, the coordinator at the source EC waits for the stream at its DF to shut down before sending a response to the coordinator at the target EC. Hence, the stream at the target EC can only be started after the stream at the source EC is shut down. This guarantees a break-before-make approach, as intended.

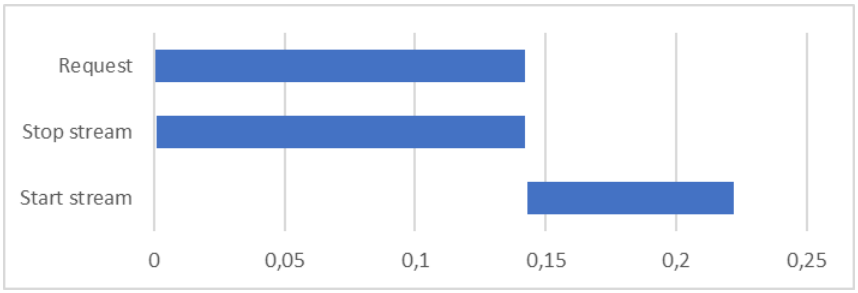

Figure 17: Timeline view of the main events related to a *BbM HO*.

Now let us take a look at the results of the *MbB HO* executions, summarized in Table 5. As we can see, the average time for stopping the stream at the old EC is 204 ms and the average time for restarting the stream at the target EC is 161 ms. From the perspective of the target EC coordinator, the average time it takes to receive a response from the source EC coordinator as depicted in Figure 13 is 53 ms. Note that for the *MbB HO*, the response to the request is sent as soon as it is ready, instead of having to wait for DF to shut down its stream.

	MbB HO 1	*MbB HO* 2	*MbB HO* 3	Average
Avg. stream stop time	256 ms	183 ms	172 ms	204 ms
Avg. stream start time	151 ms	182 ms	150 ms	161 ms
Avg. request response time	38 ms	70 ms	51 ms	53 ms

Table 5: Summary of the observed times regarding the handling of the streams for the *MbB HO* executions.

We can derive the average time for a *MbB HO* to be 214 ms. This is the result of adding the average times of getting a response to the request (53 ms) and starting the streams (161 ms). Here, it is worth mentioning that for the *MbB HO*, I consider

the process to finish when the stream is started by DF in the target EC, and not when the stream is finally closed by DF in the source EC. The reason for this is that even if the whole source EC went down after that point, it would not have any impact in terms of data delivery since the data would be sent through the stream components at the target EC. This is further discussed in section 5.1.

The main events of the *MbB HO* are illustrated in Figure 18. As we can see, the request is first served by the source EC coordinator, which allows the target EC coordinator to start the stream sooner than in the *BbM HO*. Only after the stream at the target EC is ready, the coordinator at the source EC is told to shut down its corresponding part of the stream.

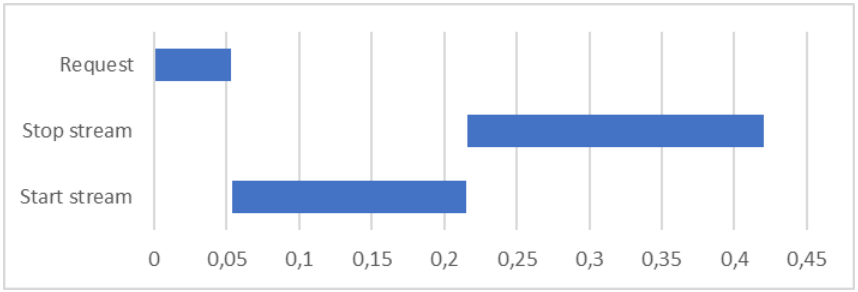

Figure 18: Timeline view of the main events related to a *MbB HO*.

By this point, I have analyzed the results regarding execution times for both solutions. We can appreciate that the total time until the stream is set in the target EC is very similar between the *BbM HO* and the *MbB HO*, with 218 ms and 214 ms respectively. However, the *MbB HO* approach guarantees no downtime of the stream, while the *BbM HO* will have the stream shutdown for 78 ms on average, corresponding to the time it takes to start the stream at the target EC in the *BbM HO*. Simulations carried by Barbera, S. et al. [2] reported HOs taking place every 2.6 seconds on average. This means that with the *BbM HO* approach, the stream will not be set roughly 3% of the time, and hence there would be a 3% chance of dropping a data point.

Now let us take a look at the implications of the results within the 5G context. Let us assume that the coverage radius of a gNB is 250 m (from Table 1). The area each gNB covers is then a circle of radius 250 m. Also let us consider the worst-case scenario, in which all UEs traverse the grid in straight lines. In particular, consider that some UE traverses the coverage area of a gNB entering at point A and exiting at point B, taking the shortest path from A to B and hence spending the least amount of time within coverage range, as depicted in Figure 19. In this setting and assuming constant speed of the UE, the shortest path corresponds to the straight segment from A to B. Each of these segments is a chord of the circle representing the coverage area.

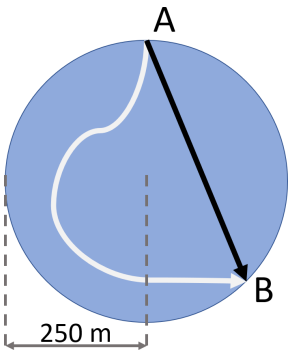

Figure 19: Visualization of the area covered by a gNB. Two possible paths for UE from point A to B are shown. In black, a chord of the circle, which is the shortest possible path and the type of paths that we will consider. In white, a longer path with a detour.

Now let us take a look at the distribution of randomly taken circle chords. Figure 20 shows the cumulative distribution function (CDF) of the length of chords on a circle of radius 250 units (such as the one we are considering for gNB coverage, meters in our case).

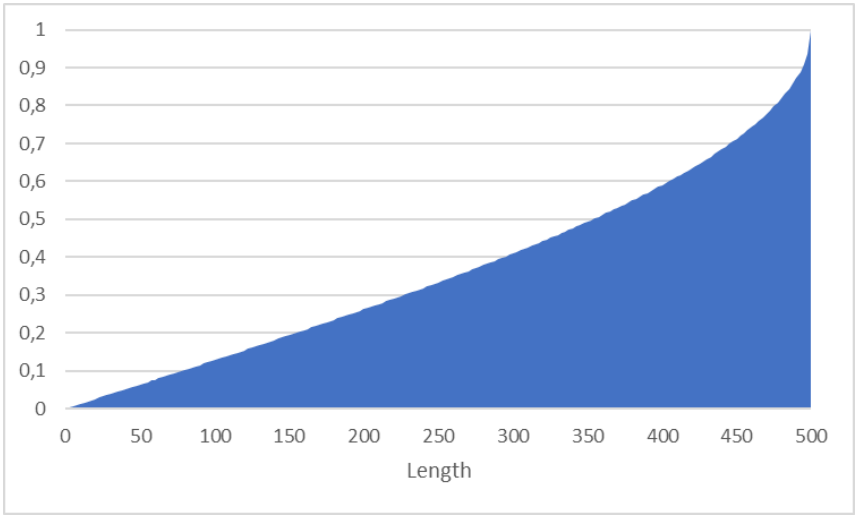

Figure 20: Cumulative distribution function of the chords of a circle of radius 250 units.

Table 6 provides the approximated length of some of the most commonly used percentiles in statistics for the aforementioned CDF. To explain what these percentiles mean, let us take percentile 75 as an example. Percentile 75 value is 462.5. This means that from all the chords of the circle of radius 250 units, 75% of them have a length equal to or less than 462.5 units. In other words, 25% of the chords will have a length greater than 462.5.

Percentile	5	10	25	50	75	90	95
Length	40	80	192.5	355	462.5	495	500

Table 6: Some of the most commonly used percentiles for the CDF pictures in Figure 20.

Now let us consider different speeds at which a mobile entity can move. In particular, I consider 8km/h (walking), 20km/h (bicycle), 50km/h (motorbike), 120km/h (car), 430km/h (high speed train), 500km/h (5G speed target [37]) and 603 km/h (highest speed registered by a train [50]).

For each of the aforementioned cases, the moving entity might traverse the gNB coverage area for such a short time that by the time the HO is complete, the UE is no longer connected to the EC that gNB provides connectivity to. I refer to this phenomenon as a vain HO or making a HO in vain. Since we have data about the different speeds, the different HO times, and the CDF of the different circle chords, we can calculate the probability of making a HO in vain for each of the aforementioned speeds. The analysis results are shown in Table 7.

			BbM HO		MbB HO	
Type	Speed (km/h)	Speed (m/s)	Average distance (m)	Probability of vain HO	Average distance (m)	Probability of vain HO
Walking	8	2.22	0.48	0.3%	0.48	0.3%
Bicycle	20	5.56	1.21	0.3%	1.19	0.3%
Motorbike	50	13.89	3.03	0.6%	2.97	0.6%
Car	120	33.33	7.26	1%	7.13	1%
Train	430	119.44	26.04	3.5%	25.56	3.5%
5G target	500	138.89	30.28	4.4%	29.72	3.8%
Train record	603	167.5	36.51	4.8%	35.84	4.8%

Table 7: Table summarizing the analysis of UE moving at different speeds w.r.t. both solutions implemented for this thesis, considering a coverage radius for the gNBs of 250m.

The first thing one might notice is that the calculated probabilities are similar for the *BbM HO* and the *MbB HO*. This is due to the HO times being similar for both, as mentioned earlier in this section. Then, regarding each moving speed, we see that for speeds up to 50km/h, the probability of making the HO in vain is below 1%. For speeds of 120km/h, the probability is nearly 1%. For the 5G maximum targeted speed, the probabilities are 4.4% for the *BbM HO* and 3.8% for the *MbB HO*. Even for higher speeds, the probabilities are still relatively low, staying below 5% for speeds up to 603km/h.

In consequence, I conclude that Nokia's system is suitable for real use and deployment. However, for the most critical cases, we must be aware of some other considerations regarding the HO procedure in the 5G setting. Note that this thesis intends to provide a base for the HO in Nokia's system, so the following are remarks for further work and are out of the scope of this thesis.

In section 2.1 I talk about how cellular networks are often depicted as hexagonal grids. While this is valid to get an understanding of how cellular networks operate, this is too much of a simplification for the 5G setting. As I described in section 2.2, 5G considers cell types with different characteristics. Hence, a more realistic view of the 5G cellular network is depicted in Figure 21. Grey circles represent the coverage areas of different 5G cells to which a particular UE might connect along the path indicated by the black arrow. As we can appreciate, the topology of the network is complex. In real use cases, we will want Nokia's system to avoid performing a HO to cells with short coverage ranges, since we can expect the UE to spend little time in those, resulting in high probabilities of making a HO in vain. The system could be optimized by providing it with information about the network topology and UE path prediction capabilities. Combined, they would enable faster HOs by preparing the components at the candidate target ECs in advance.

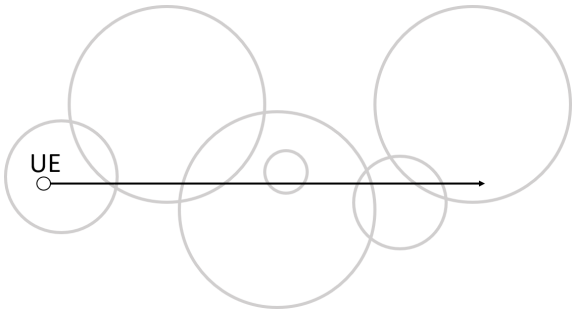

Figure 21: Example of a mobile UE path in the 5G cellular network. Circles represent coverage areas of ECs to which the UE might connect.

Now let me discuss another important aspect of HO in real-life scenarios. Consider the scenario depicted in Figure 22, consisting of a UE present at some source EC and which is about to perform a HO to a target EC. In that scenario, there will be a time A at which the UE will establish a connection to the target EC, a time B at which connection to the source EC will be lost, and a time C at which the stream will be ready at the target EC.

We can make three distinctions depending on the timing of C:

- C happens between A and B. This would be the ideal case, in which the stream would not be interrupted at all since the stream would be ready at the target EC once the UE leaves the source EC.

- C happens between B and the moment when the UE loses connectivity with the target EC. This case is depicted in Figure 22. Note that there will be a time window between B and C in which the UE will not be able to send any data due to no connection being established to any EC. In some cases, it can be acceptable to drop data points in that case, avoiding any overhead. However, for critical cases, if no data can be dropped, the UE must be able to buffer the data points until a connection is established to the target EC at some point. The buffer should provide the capability to store as many data points as

possible, hence reducing the probability of data being dropped. Further details are out of the scope of this thesis.

- C happens after the UE has lost connectivity to the target EC. This would be the case of the HO taking place in vain. After a HO in vain, the UE will lose connectivity with Nokia's system and is not a trivial problem how to orchestrate the system to provide access to the system in future ECs in a timely manner. My proposed approach is to employ path prediction technologies to make sure the system is ready in the different ECs even before the UE notices them, avoiding this problem entirely. However, this is left as further research.

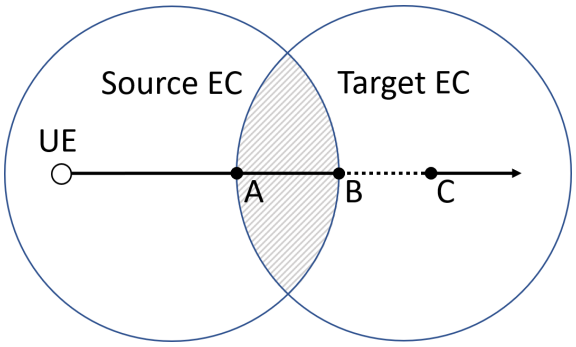

Figure 22: Scenario in which a UE will HO from the source EC to a target EC.

4.3.2 Other differences

I have discussed the different solutions from the perspective of stream management overhead and explained its implications on theoretical scenarios. There are other significant differences that are worth mentioning too.

The first one has been mentioned previously in this thesis, but cannot be overstated, and is what separates one solution from the other. In the *MbB HO*, the HO is done through a make-before-break approach, which is enabled by 5G's multi-connectivity [26], meaning that UE will be able to simultaneously connect to several gNBs. This was unfeasible in previous network generations due to technical limitations. The main benefit is that it will allow for data-lossless mobility, or in other words, mobility without connectivity breaks.

Another difference is that, from the HO perspective, for the *MbB HO* approach there is no difference regarding whether DF was pre-deployed in the target EC, under the assumption that the connection to the source EC is not lost during the deployment of DF. As explained at the beginning of this chapter, for my testing environment, deploying DF implied an overhead of 10 seconds. For the *BbM HO*, this is important since we are stopping the stream at the source EC before creating the new stream at the target EC. This means that if DF is not present when a *BbM HO* takes place, there will be a 10-second interval in which the stream will not be up. Hence, to minimize its impact on real case scenarios, each EC should launch DF along with coordinator during the startup of Nokia's system. However, as mentioned

earlier, for the *MbB HO*, there is no difference w.r.t. the presence of DF in the target edge, since the stream at the source EC will be used until the new stream is set-up (assuming that the UE is still in the range of the source EC gNBs coverage), regardless of the time DF takes to initialize.

Another difference between the solutions is that, in the MbB HO, DH might appreciate a spike in network traffic received from a stream that is going through the HO procedure. More precisely, the traffic regarding such a stream could double for some time. This effect is due to data possibly being sent from two different DFs, one running at the source EC and the other at the target EC. This does not necessarily happen on all MbB HOs. The time for which DH might be affected by this depends on the communication delays between the two ECs and the time it takes for the source EC to shut down its stream. I discuss this phenomenon and its implications further in section 5.1. Note that in BbM HO this will not happen since the system guarantees that the DF at the source EC stops its stream before any information is sent to the coordinator at the target EC.

4.4 Conclusion

The obtained results imply that Nokia's system is capable of handling UE mobility in the 5G context. By enabling make-before-break HO, the system should be ready for use cases requiring the least amount of delay in the responses, such as self-driving cars. However, network delays in the messaging of the different KPI values were not measured due to not being able to properly emulate the 5G cellular network in the considered testing environment and is left for further research.

5 Further work

In this thesis, I explored a first implementation of data processing HO in the 5G and MEC setting for Nokia's system. However, during the ideation and implementation of the different solutions, I came across several areas of further research and discussion which should not be overlooked. In this section, I will discuss the most interesting and relevant ones. Note that all of the following are left as further work since they are complex problems that are out of the scope of this thesis.

5.1 Stream synchronization

Let us consider the scenario of the *MbB HO* taking place, as explained in section 3.3.2. Furthermore, let us consider we are at the point in which the new stream has been created for the DF at the target EC, but the stream at the target EC has still not been shut down.

The following components would then be present in Nokia's system:

- A single data source, generating some data. For this example, let us consider that it is generating readings of two variables of interest every second.
- Two DFs. Let us call them DF1 and DF2. We can consider them to be on different ECs. Both of them should ideally produce the same KPI values. However, as we will see, this is not necessarily the case. For this example, let us consider that they are set to calculate the KPI values every three seconds. Also, for simplicity, let us consider that the KPI values are just the average of each of the readings over each time window.
- A DH, which will receive the KPI values produced by both DF1 and DF2. At some point, it will stop receiving data from DF1, even though this does not influence the problem.

As mentioned above, in ideal circumstances, the KPI values produced by both DF1 and DF2 should be the same. However, in practice, we will probably experience delays in the data delivery due to various reasons, the main ones being delays in the delivery of the messages due to network latency issues and delays regarding the programmed timers, as Java timers do not offer real-time guarantees [71].

To explain the problem at hand, let us consider that a data point with timestamp 12 is processed in time by DF2, hence taking it into account when calculating the averages, while DF1 is not able to process it for the same time window due to, for example, some delay in the network. Note that DF1 will process that time on the next time window.

This will result in a mismatch between the KPI values produced by the different DFs. The scenario is depicted in Figure 23.

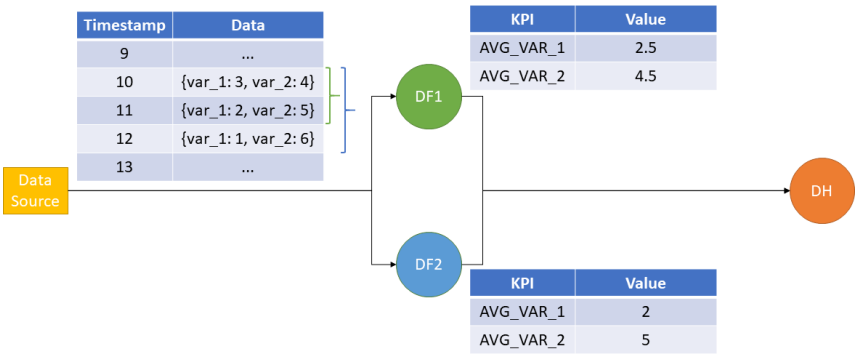

Figure 23: Scenario showing how a mismatch between KPI values coming from different DFs could happen.

Let us now look at the situation from the perspective of DH, whose perspective is depicted in Figure 24. DH will receive two messages, one from each DF, containing data about the KPI values. Let us say it first receives the message from DF1. The message will be forwarded to the appropriate subscribers, as usual. However, once the message from DF2 is received, DH will notice, by looking at the timestamps, that the KPI values involved in said message have overlapped with a previous message.

In the current implementation of the solution, this is not treated as a special case and DH will simply forward the data to the appropriate subscribers. However, this can be problematic. If the data we are dealing with happens to be sensitive, such that it triggers an alarm, the alarm might be triggered more than once. Another scenario would be that of self-driving cars. In this case, the car might be given instructions to steer more than what it really should. Overlapping data can be fatal for critical systems if not detected and handled properly.

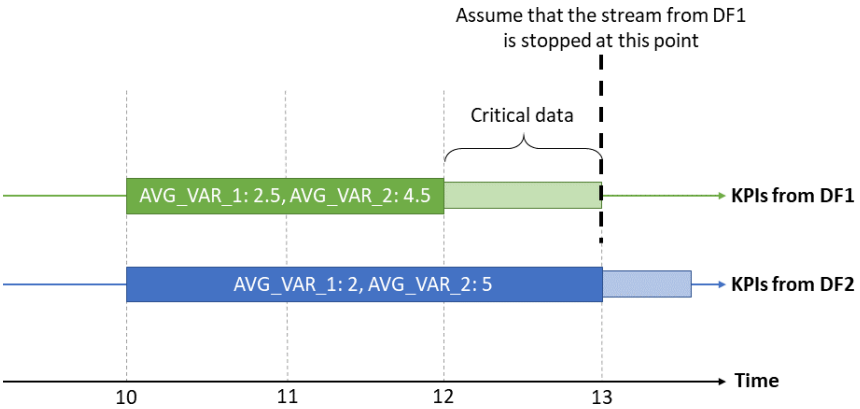

Figure 24: Timeline depicting the stream synchronization problem from DH's perspective.

In Figure 24, we can appreciate that the data belonging to the time window from 12 to 13 will not be properly handled by the system. In particular, if the stream from DF1 is shutdown on instant 13, no further data points will be received by DH from DF1. Then, since the last message from DF1 was already forwarded to the

appropriate subscribers by the time of receiving the message from DF2, the system will have two options:

- Forward the message from DF2. This will result in forwarding a KPI encoding data which was, in some way, already forwarded. As mentioned earlier, this can have fatal consequences.
- Do not forward the message from DF2. This would inevitably result in no data from the time window from 12 to 13 being delivered to the subscribers, assuming that the next message starting from instant 13 from DF2 would be delivered normally. This can also result in fatal consequences.

Hence, it is important to solve this problem for Nokia's system to enable a fully reliable make-before-break data processing HO on the 5G cellular network context.

5.2 More realistic implementation and testing

In chapter 4 I discussed the theoretical results of the solution implemented for this thesis. The testing environment depicted there is far from realistic, despite being valid to confirm the usability of the solution. Hence, in this section, I describe different steps or approaches that could be taken to take testing to the next phase.

First, let me talk about how the HO is being emulated. As mentioned in section 4.2, the HO emulation was done by spawning a DF in the target EC of a HO. That way, the data source can be noticed by the coordinator of said EC and the HO process can start. In real-life scenarios, this translates to requiring a pre-deployed DF in every EC. However, from Nokia's system logic perspective, said pre-deployed DF should not be necessary, and Nokia's system should be able to spawn it on demand, hence allowing resource optimization at the EC level. This would inevitably require the detection of the new data source by the coordinator of the target EC, rather than it being DF's responsibility. However, we must notice that this would create a trade-off between resource optimization and HO speed, due to having to wait for DF to spawn to allow for the HO to start.

Another interesting area of research for further work is to analyze and compare the behavior of the system in a real scenario. For the evaluation of this thesis, the HO was as isolated as possible within the system and from the 5G and MEC setting, to enable a theoretical analysis of the system capabilities. After confirming its suitability for the 5G environment, the next logical step would be to analyze its performance on a real use case, in which the system must handle many more events apart from the HOs. Not only that, but 5G cellular network emulation tools could also be used for said purpose.

5.3 Resource cleanup after handover

As we saw in section 2.2.4, resources at the EC will be limited, and they will also be shared by all running applications within the EC. Hence, it is important that applications make efficient use of the resources to provide a good user experience.

For the matters of this thesis, in sections 3.3.1 and 3.3.2 I explained how the stream template model is sent from the source EC to the target EC and stored there in the case of a HO. However, in the current state, an EC will keep storing the stream template models even if there are no present data sources anymore. In the 5G context, we expect a high number of UE traversing the grid, each of them having its template model. This might result in an EC running out of resources to store new stream template models. This will result in failing to provide a good QoS, both from the user and the application perspectives.

Hence, this is an issue that must be taken care of. A first basic approach would be to simply delete the stream template model within an EC when all the data sources which required it have shut down the connections to said EC. However, we must be aware of two things. First, this would add another layer of complexity to keeping track of the different data sources, since some of them might lose connection temporarily or enter energy-saving mode, hence also shutting down the connection for some time. Second, that approach brings us a trade-off between memory usage at the EC local data center and network usage. Network usage would be caused due to having to fetch the stream template back if the data source was to be noticed again. Hence, the solution to this problem should be taken by analyzing each use case separately.

As a final note regarding stream template models, for the work of this thesis, it is assumed that a data source is connected to some EC and the stream template model is available to the coordinator of said EC from the beginning, as can be seen in Figure 6. However, in future revisions of Nokia's system, this will not be the case and the stream template model will need to be fetched from some online server or central repository. Hence, another area of research would be to study where the stream template models should be and how to handle said initial fetching.

6 Summary

In this thesis, I study two implementations of data processing HO within Nokia's system. One emulates older technologies which only allowed for a break-before-make HO solution. That solution, which I named *break-before-make (BbM) HO*, is presented only to be used as a baseline to which to compare the actual solution. The second solution, the *make-before-break (MbB) HO*, takes advantage of newer revisions of the cellular network which allow for make-before-break solutions, among other improvements. For the context of this thesis, this means that the data processing HO from the source EC to the target EC will be carried without the data streams being interrupted. This sets the ground for data lossless HOs.

I also provide an analysis of the feasibility of both solutions in the 5G context. The analysis was carried with information coming directly from the execution logs of Nokia's system. Both solutions are reported as technically suitable to be used in real use cases. However, for critical use cases that cannot accept any data points not being delivered, the *MbB HO* is preferred.

While the *MbB HO* is the first step towards a fully functional data processing HO mechanism for the 5G and MEC setting, there are still some details which need attention before it can be considered final. Among those, I highlight the problem of stream synchronization, explained in section 5.1.

References

[1] N. Bhandari, S. Devra and K. Singh. Evolution of Cellular Network: From 1G to 5G. *International Journal of Engineering and Techniques,* 2017, vol. 3, no. 5, pp. 98-105.

[2] S. Barbera et al. Synchronized RACH-less handover solution for LTE heterogeneous networks. *2015 International Symposium on Wireless Communication Systems (ISWCS), Brussels,* 2015, pp. 755-759.

[3] A. Ghosh, V. V. Paranthaman, G. Mapp, O. Gemikonakli and J. Loo. Enabling seamless V2I communications: toward developing cooperative automotive applications in VANET systems. *IEEE Communications Magazine,* 2015, vol. 53, no. 12, pp. 80-86.

[4] S. Chen, J. Hu, Y. Shi and L. Zhao. LTE-V: A TD-LTE-Based V2X Solution for Future Vehicular Network. *IEEE Internet of Things Journal,* 2016, vol. 3, no. 6, pp. 997-1005.

[5] M. T. Beck, M. Werner, S. Feld and T. Schimper. Mobile Edge Computing: A Taxonomy. *AFIN 2014: The Sixth International Conference on Advances in Future Internet,* 2014, pp. 48-54.

[6] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis and A. Vakali. Cloud Computing: Distributed Internet Computing for IT and Scientific Research. *IEEE Internet Computing,* 2009, vol. 13, no. 5, pp. 10-13.

[7] Y. Jadeja and K. Modi. Cloud computing - concepts, architecture and challenges. *2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET), Kumaracoil,* 2012, pp. 877-880.

[8] M. Lauridsen, L. C. Gimenez, I. Rodriguez, T. B. Sorensen and P. Mogensen. From LTE to 5G for Connected Mobility. *IEEE Communications Magazine,* 2017, vol. 55, no. 3, pp. 156-162.

[9] M. Shafi et al. 5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice. *IEEE Journal on Selected Areas in Communications*, 2017, vol. 35, no. 6, pp. 1201-1221.

[10] I. Grigorik. High Performance Browser Networking. *O'Reilly Media, Inc,* Online book, pp. Mobile Networks. URL: https://hpbn.co/mobile-networks/

[11] S. Kapoor. Mobile Edge Compression of Management Data in Cellular Networks using Quality of Monitoring Classes. Master's Thesis. *University of Helsinki, Helsinki,* 2017. Accessed 5.7.2019. URL: https://helda.helsinki.fi/handle/10138/228837

[12] L. Hobert, A. Festag, I. Llatser, L. Altomare, F. Visintainer and A. Kovacs. Enhancements of V2X communication in support of cooperative autonomous driving. *IEEE Communications Magazine*, 2015, vol. 53, no. 12, pp. 64-70.

[13] M. Olsson and C. Mulligan. EPC and 4G Packet Networks: Driving the Mobile Broadband Revolution. *Academic Press, Amsterdam,* 2012.

[14] A. Zavodovski, N. Mohan, S. Bayhan, W. Wong, and J. Kangasharju. ICON: Intelligent Container Overlays. *Proceedings of the 17th ACM Workshop on Hot Topics in Networks. ACM,* 2018, pp. 15-21.

[15] P. Domingos. A Few Useful Things to Know About Machine Learning. *Communications of the ACM,* 2012, vol. 55, no. 10, pp. 78-87.

[16] S. Chen et al. Vehicle-to-Everything (v2x) Services Supported by LTE-Based Systems and 5G. *IEEE Communications Standards Magazine,* 2017, vol. 1, no. 2, pp. 70-76.

[17] M. M. Zanjireh and H. Larijani. A Survey on Centralised and Distributed Clustering Routing Algorithms for WSNs. *2015 IEEE 81st Vehicular Technology Conference (VTC Spring), Glasgow,* 2015, pp. 1-6.

[18] A. Prakash, S. Tripathi, R. Verma, N. Tyagi, R. Tripathi and K. Naik. Vehicle assisted cross-layer handover scheme in NEMO-based VANETs (VANEMO). *International Journal of Internet Protocol Technology (IJIPT),* 2011, vol. 6, no. 1/2.

[19] R. Molina-Masegosa and J. Gozalvez. LTE-V for Sidelink 5G V2X Vehicular Communications: A New 5G Technology for Short-Range Vehicle-to-Everything Communications. *IEEE Vehicular Technology Magazine,* 2017, vol. 12, no. 4, pp. 30-39.

[20] B. Toghi et al. Multiple Access in Cellular V2X: Performance Analysis in Highly Congested Vehicular Networks. *2018 IEEE Vehicular Networking Conference (VNC), Taipei, Taiwan,* 2018, pp. 1-8.

[21] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. *Final draft ETSI EN 302 637-2 V1.3.1 (2014-09).* URL: https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.03.01_30/en_30263702v010301v.pdf

[22] Z. Xu, X. Li, X. Zhao, Zhang, M. H. and Wang, Z. DSRC versus 4G-LTE for Connected Vehicle Applications: A Study on Field Experiments of Vehicular Communication Performance. *Journal of Advanced Transportation,* 2017, vol. 2017, Article ID 2750452, 10 pages.

[23] V. Kojola, S. Kapoor, K. Hätönen. Distributed Computing of Management Data in a Telecommunications Network. *Mobile Networks and Management (MONAMI),* 2016, pp. 146-159.

[24] S. S. Babatola. Global burden of diseases attributable to air pollution. *Journal of public health in Africa,* 2018, vol. 9, no. 813, pp. 162-166.

[25] D. Schwela et al. Urban Air Pollution in Asian Cities. *Earthscan,* 2006, pp. 91, 212.

[26] M. Giordani, M. Mezzavilla, S. Rangan and M. Zorzi, Multi-connectivity in 5G mmWave cellular networks. *2016 Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), Vilanova i la Geltru,* 2016, pp. 1-7.

[27] Nokia. Machine Learning in 5G Networks for Enhanced Mobile Broadband and Industry Solutions. Whitepaper. Online resource. Accessed 2.7.2019. URL: https://onestore.nokia.com/asset/206223

[28] Nokia. 5G new radio network: Use cases, spectrum, technologies and architecture. Whitepaper. Online resource. Accessed 2.7.2019. URL: https://onestore.nokia.com/asset/205407

[29] Nokia. Unleashing the economic potential of network slicing. Whitepaper. Online resource. Accessed 2.7.2019. URL: https://onestore.nokia.com/asset/202089

[30] Nokia. Low latency in 4.9G/5G: Solutions for millisecond latency. Whitepaper. Online resource. Accessed 2.7.2019. URL: https://onestore.nokia.com/asset/201407

[31] Rohde & Schwarz. Narrowband Internet of Things. Whitepaper. Online resource. Accessed 26.7.2019. URL: https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/application_notes/1ma266/1MA266_0e_NB_IoT.pdf

[32] ETSI. Mobile Edge Computing - A key technology towards 5G. Whitepaper. Online resource. Accessed 5.7.2019. URL: https://yucianga.info/wp-content/uploads/2015/11/Ref02-2015-09-etsi_wp11_mec_a_key_technology_towards_5g.pdf

[33] Nokia. Dynamic end-to-end network slicing for 5G. Whitepaper. Online resource. Accessed 9.7.2019. URL: https://onestore.nokia.com/asset/200339

[34] Interxion. Truth and lies about latency in the cloud. Whitepaper. Online resource. Accessed 5.7.2019. URL: https://www.interxion.com/globalassets/_documents/whitepapers-and-pdfs/cloud/WP_TRUTHANDLIES_en_0715.pdf

[35] K. Lasse Lueth. State of the IoT 2018: Number of IoT devices now at 7B - Market accelerating. Online resource. Accessed 29.3.2019. URL: https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/

[36] IGI Global. What is Cellular Network. Online resource. Accessed 2.4.2019. URL: https://www.igi-global.com/dictionary/cellular-network/3547

[37] GSMA. Road to 5g: Introduction and Migration. Online resource. Accessed 5.7.2019. URL: https://www.gsma.com/futurenetworks/wp-content/uploads/2018/04/Road-to-5G-Introduction-and-Migration_FINAL.pdf

[38] RF Wireless World. 5G speed vs 5G range-What is the value of 5G speed,5G range. Online resource. Accessed 26.7.2019. URL: https://www.rfwireless-world.com/Terminology/5G-Speed-Vs-5G-Range.html

[39] Telecom ABC. Cellular Network. Online resource. Accessed 2.4.2019. URL: http://www.telecomabc.com/c/cellular.html

[40] S. Kavanagh. How fast is 5G?. Online resource. Accessed 1.4.2019. URL: https://5g.co.uk/guides/how-fast-is-5g/

[41] Mobile Network Guide. Mobile Base Stations. Online resource. Accessed 1.4.2019. URL: https://www.mobilenetworkguide.com.au/mobile_base_stations.html

[42] Opensignal. State of Mobile Networks: USA (February 2017). Online resource. Accessed 1.4.2019. URL: https://www.opensignal.com/reports/2017/02/usa/state-of-the-mobile-network

[43] A. Nordrum et al. Everything You Need to Know About 5G. Online resource. Accessed 1.4.2019. URL: https://spectrum.ieee.org/video/telecom/wireless/everything-you-need-to-know-about-5g

[44] Nokia. 5G Deliver the extraodrinary. Online resource. Accessed 2.7.2019. URL: https://www.nokia.com/networks/5g/

[45] JetBrains. IntelliJ IDEA. Online resource. Accessed 25.7.2019. URL: https://www.jetbrains.com/idea/

[46] Apache. Kafka. Online resource. Accessed 25.7.2019. URL: https://kafka.apache.org/

[47] M. Veeraraghavan. Three planes in networks. Online resource. Online resource. Accessed 3.4.2019. URL: http://www.ece.virginia.edu/mv/edu/ee136/Lectures/routing-sig/cs-ps-cops.pdf

[48] 3GPP. About 3GPP. Online resource. Accessed 8.4.2019. URL: https://www.3gpp.org/about-3gpp

[49] Multi-Tech Systems, Inc. Anticipated Cellular Carriers 2G/3G Sunset Dates. Online resource. Accessed 8.4.2019. URL: https://www.multitech.com/documents/publications/marketing-guides/MT_Anticipated_Sunset_Cellular_Carriers_PDF.pdf

[50] The Guardian. Japan's maglev train breaks world speed record with 600km/h test run. Online resource. Accessed 4.7.2019. URL: https://www.theguardian.com/world/2015/apr/21/japans-maglev-train-notches-up-new-world-speed-record-in-test-run

[51] WorldTimeZone. 4G map LTE World Coverage Map - LTE WiMAX HSPA 3G GSM Country List. Online resource. Accessed 2.7.2019. URL: https://www.worldtimezone.com/4g.html

[52] R. Meakin, S. Wong, K. Zikry and D. Shea. Making 5G pay: Monetizing the impending revolution in communications infrastructure. Online resource. Accessed 18.7.2019. URL: https://www.strategyand.pwc.com/report/Making-5G-pay

[53] University of Helsinki. Air quality monitoring. Online resource. Accessed 11.7.2019. URL: https://www.helsinki.fi/en/researchgroups/sensing-and-analytics-of-air-quality/topics/air-quality-monitoring

[54] Vaisala. Air Quality Transmitter AQT400 Series. Online resource. Accessed 11.7.2019. URL: https://www.vaisala.com/sites/default/files/documents/AQT400-Series-Datasheet-B211581EN_0.pdf

[55] South Coast Air Quality Management District. Field Evaluation of Vaisala Air Quality Transmitter AQT410. Online resource. Accessed 11.7.2019. URL: http://www.aqmd.gov/docs/default-source/aq-spec/field-evaluations/vaisala---field-evaluation.pdf?sfvrsn=10

[56] M. Meriläinen-Tenhu. University of Helsinki and Nokia Bell Labs develop smart 5G technology to monitor air quality. Online resource. Accessed 4.4.2019. URL: https://www.helsinki.fi/en/news/data-science-news/university-of-helsinki-and-nokia-bell-labs-develop-smart-5g-technology-to-monitor-air-quality

[57] University of Helsinki. About Megasense. Online resource. Accessed 29.3.2019. URL: https://www.helsinki.fi/en/researchgroups/sensing-and-analytics-of-air-quality/about-Megasense

[58] World Health Organization. How air pollution is destroying our health. Online resource. Accessed 11.7.2019. URL: https://www.who.int/air-pollution/news-and-events/how-air-pollution-is-destroying-our-health

[59] Vaisala. Let's Make Our Cities Breathable. Online resource. Accessed 11.7.2019. URL: https://www.vaisala.com/sites/default/files/documents/WEA-MET-G%20Air%20Quality%20Brochure_B211714EN-A.pdf.pdf

[60] DDS Foundation. Why Choose DDS?. Online resource. Accessed 24.7.2019. URL: https://www.dds-foundation.org/why-choose-dds/

[61] NASA. New Map Offers a Global View of Health-Sapping Air Pollution. Online resource. Accessed 11.7.2019. URL: https://www.nasa.gov/topics/earth/features/health-sapping.html

[62] Beijing Municipal Bureau of Statistics. Economic Development of Beijing Maintained a Stable and Good Momentum in 2017. Online resource. Accessed 11.7.2019. URL: http://tjj.beijing.gov.cn/English/PR/201801/t20180125_391609.html

[63] University of Helsinki. Research - Sensing and Analytics of Air Quality. Online resource. Accessed 11.7.2019. URL: https://www.helsinki.fi/en/researchgroups/sensing-and-analytics-of-air-quality/research

[64] J. Horwitz. South Korea hits 1 million 5G subscribers in 69 days, beating 4G record. Online resource. Accessed 2.7.2019. URL: https://venturebeat.com/2019/06/12/south-korea-hits-1-million-5g-subscribers-in-69-days-beating-4g-record/

[65] EventHelix. LTE Random Access Procedure. Online resource. Accessed 25.7.2019. URL: https://www.eventhelix.com/lte/random-access-procedure/lte-random-access-procedure.pdf

[66] M. Burns. Verizon's 5G to launch first in Chicago and Minneapolis on April 11. Online resource. Accessed 2.7.2019. URL: https://techcrunch.com/2019/03/13/verizons-5g-to-launch-first-in-chicago-and-minneapolis-on-april-11/

[67] GSMA. LTE and 5G Market Statistics – 8 April 2019. Online resource. Accessed 19.7.2019. URL: https://gsacom.com/paper/lte-5g-market-statistics-8-april-2019/?utm=reports4g

[68] TRTWorld. South Korea launches first national 5G networks. Online resource. Accessed 2.7.2019. URL: https://www.trtworld.com/business/south-korea-launches-first-national-5g-networks-25563

[69] J. Elliott. On the road to 5G use cases. Online resource. Accessed 2.7.2019. URL: https://www.nokia.com/blog/road-5g-use-cases/

[70] ETSI. ETSI publishes European Standards for Intelligent Transport Systems. Online resource. Accessed 17.11.2019. URL: https://www.etsi.org/newsroom/news/851-2014-12-press-etsi-publishes-european-standards-for-intelligent-transport-systems?jjj=1563270570671

[71] Oracle. Timer (Java Platform SE 8). Online resource. Accessed 5.7.2019. URL: https://docs.oracle.com/javase/8/docs/api/java/util/Timer.html